

PETTM Fun and Games

Selected  Programs

Ron Jeffries and
Glen Fisher



PETTM Fun and Games

Selected **CURSORTM** Programs

Ron Jeffries and
Glen Fisher



Disclaimer of Warranties and Limitation of Liabilities

The authors have taken due care in preparing this book and the programs in it, including research, development, and testing to ascertain their effectiveness. The authors and the publishers make no expressed or implied warranty of any kind with regard to these programs nor the supplementary documentation in this book. In no event shall the authors or the publishers be liable for incidental or consequential damages in connection with or arising out of the furnishing, performance, or use of any of these programs.

CURSOR™ is a trademark of The Code Works.

PET and **CBM** are trademarks of Commodore Business Machines, Inc. *PET Fun and Games* is not sponsored or approved by or connected with Commodore Business Machines, Inc. All references to **PET** and **CBM** in the text of this book are to the trademarks of Commodore Business Machines, Inc.

Master Mind® is a registered trademark of Invitica Plastics.

Othello® is a registered trademark of Gabriel Industries Inc.

Yahtzee® is a registered trademark of E.S. Lowe Company Inc.

Chutes and Ladders® is a registered trademark of Milton Bradley Company.

**Published by
OSBORNE/McGraw-Hill
630 Bancroft Way
Berkeley, California 94710
U.S.A.**

For information on translations and book distributors outside of the U.S.A., please write
OSBORNE/McGraw-Hill at the above address.

PET™ FUN AND GAMES

Copyright © 1981 by McGraw-Hill, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

1234567890 HCHC 87654321

ISBN 0-931988-70-5

Cover art and text illustrations by Jenni Jeffries

Cover concept by KLT van Genderen

ACKNOWLEDGMENTS

We want to thank all of the people who have helped make CURSOR and this book possible. Most important are the authors of the programs. This book is a tribute to their creativity and imagination.

Ralph McElroy deserves special mention, since he is the one who invented the "cassette magazine" idea when he introduced CLOAD for the TRS-80 in early 1978. One of us (Ron) tried to talk Ralph out of what seemed like a crazy idea, but he ignored our advice, and CLOAD became an overnight success story. When we decided to start CURSOR as a cassette magazine for the PET, Ralph was very helpful.

We want to thank all of the people who have subscribed to CURSOR for their support. Obviously, without subscribers, CURSOR wouldn't exist and this anthology of programs would not have been written. But more than that, they've been kind enough to tell us about our mistakes, and always have been a rich source of ideas for programs.

The entire Jeffries family has been part of CURSOR since its inception. Jann Jeffries has kept the books straight, and assisted in almost

every phase of the business. Scott and Jennifer have helped with everything from duplicating tapes to stuffing envelopes.

Bob Albrecht has been a friend and supporter of CURSOR since the early days. Dana Trout, Gregg Pickles, and Stan Reifel have helped us solve all sorts of hardware problems. Nancy Rhodes played a major role in helping prepare this book for publication. She got us organized, contacted program contributors, and supervised much of the program testing.

All of the following people have tested CURSOR programs and deserve credit for finding obscure bugs, and helping to polish the programs: Mike Rowley, Todd Zimmerman, Eric Friesen, Diane Grove, Brian Sawyer, Shawn Meehan, Dave Iwanaga, Andy Stadler, Randi Opp, and Scott Jeffries.

And finally, a special thanks to the great folks at OSBORNE/McGraw-Hill. They've been a delight to work with.

R. J.
G. F.

CONTENTS

ACTION GAMES

ZAP 2

Maneuver your “zapper” around an arena and score points by “zapping” targets.

SHARK 6

Emerging from the depths of the ocean, you are the “jaws” just looking for a meal.

DEMON 13

A demon has escaped during a public demonstration and your job is to capture it.

FERRY 17

Deep in outer space, your mission is to ferry provisions to a stranded expedition.

BAT 21

You are a bat, and you’re very hungry; unfortunately, you have a few handicaps that hinder your eating habits.

NAB 26

It’s off to the races, but the track is littered with money as well as ruthless opponents.

PUZZLES

HANOI 32

If a group of Southeast Asian priests can figure out how to move the golden disks, so can you.

BOX 37

Join the electronics geniuses of Silicon Valley in producing circuit chips for microcomputers.

MIND 43

Try to outwit the PET by guessing the computer’s colors and patterns for a set of pegs.

PEG 49

It’s apparently an easy task to switch two sets of pegs, but a few twists add a bit more challenge.

LEAP 53

Remove pegs one by one simply by leaping over them, but try to end up with just one peg and you’ll discover the difficulty.

FIFTEEN 57

Slide the tiles around the board, but be sure that you finish your juggling with the tiles in numerical order.

RATRUN 60

Now's your chance to wander through a maze in search of cheese, only here it is done from a "rat's eye" view.

GAMES OF RISK

MAZE 70

The PET has constructed a maze filled with gold, but your mission is to find the gold without the aid of a map.

DUNGEON 75

Deep within some dark and dank dungeons, you must outwit some ancient monsters in order to get their gold.

EVEREST 80

Watch your step in this one, as the harsh elements of the world's tallest mountain hinder your trek to the peak.

GODZILLA 85

One of the world's most famous monsters is on the rampage again, and you must defeat it without the help of any giant apes.

FIRE 91

The inferno has started and only you can prevent total disaster, but you must be quick to avoid a pile of smoldering ashes.

GAMES OF STRATEGY

DOTS 98

You may have thought you had played everyone in this favorite game, but you haven't lived until you've played the PET.

REVERSI 103

The game may be as old as time itself, but just try to defeat the PET with strategy and guile.

PIEGRAM 109

Look out because the PET is flinging pies your way, but the catch is that it is invisible while doing so.

COPS 115

It's the old cops and robbers chase routine through the city with some very clever crooks to be chased.

SPOT 120

You only need four squares in a row to win, but the fun is in how you place the squares to get your winning combination.

GAMES OF CHANCE

BJACK 126

Deal another hand because this imaginary card playing game allows you all the money you can spend.

YAHTZEE 133

If you think rolling five dice to come up all with the same number is easy, perhaps you should try this game.

BONZO 140

Play against a friend or against the computer, but this is a race to the top with a few obstacles thrown in.

BOSWAIN 144

Pay attention to this guy's hands because he certainly has a hungry eye on yours.

GAMES FOR FUN

MAD 152

Composing stories may not have been your idea of a good time before, but with the aid of a computer it can be most interesting.

QUIX 156

Remember what you just saw and heard and be able to repeat it exactly, or you'll never defeat the PET.

BOP 160

If you think counting on your fingers is old-fashioned, perhaps you'll reconsider after playing this game.

FACE 167

Your artistic talents can now bubble forth through the screen of the PET as you compose a variety of mugs.

INTRODUCTION

P*ET Fun and Games* provides you with complete listings for 31 playful programs written specifically for the Commodore PET and CBM personal computers. The programs take full advantage of special features of these Commodore computers, including their unique character graphics. All of the programs are written in BASIC, but you don't need to be a programmer to play these games; just enter them from the listings provided.

All of the programs included here have a common goal: to be fun! Some of the programs will test your memory, some will test your strategy, others will challenge your creativity, and still others will offer you some educational value. In some games you may play against another person, but beware, because in others your opponent is the computer.

Some readers may already know of a unique magazine named *CURSOR*. It is unique in part because it publishes programs only for Commodore PET and CBM computers. It is also unique because it publishes the programs on cassette. Subscribers then "read" the magazine by loading the programs into the PETs and CBMs.

The programs included in *PET Fun and Games* have been carefully selected from all the programs published in *CURSOR* over the past three years. Fourteen programmers have contributed to this book, each with individual style and approach. With that, we offer here some of the best, most showy, entertaining and fun programs ever written for the Commodore computers.

How to Use This Book

Complete listings are provided for each program. This means you can type the program in your computer exactly as it is listed here, and then play! The instructions on how to play each game are provided within each program, with supplemental instruction offered in the introduction to each program.

The programs make extensive use of the graphic characters standard to the Commodore PET computers. These graphic characters are part of the standard character set of every PET mode;

you can see the characters on each key on the keyboard. The CBM computer does not normally display graphic characters; instead, it normally displays upper- and lower-case alphabetic characters. If you are using a CBM, you can enable most graphics characters by typing **POKE 59468,12** before entering the program.¹ (After entering the program, you can type **POKE 59468,14** to return the CBM to its normal upper- and lower-case alphabetic character set.) Some graphics cannot be typed on a CBM with a business keyboard. You will have to enlist the aid of a friend with a PET to enter them.

Whenever you enter any program, be sure to save it on a cassette or diskette so you won't have to type it in again. If you own a CBM but have access to a PET, you can also enter the programs using a PET, and then run them on your CBM.

All of the programs are also available on cassette or diskette from the folks at *CURSOR*. For information, you should write to:

PET FUN AND GAMES
c/o *CURSOR* Magazine
5266 Hollister, Suite 224
Santa Barbara, CA 93111
U. S. A.

How to Read the Listings in This Book

We have used special notation in the listings in order to make them easier to read. You should become familiar with this notation before you begin to enter any listing.

First and foremost, anything printed within curly braces is special notation. Curly braces look like this: { }. Anything printed within these braces, including the braces themselves, should not be keyed into your listing as you see it. All special notation is explained in the following paragraphs.

You should, however, *enter every character not printed within a set of curly braces exactly as you see it printed in the listing, with one exception: spaces.*

¹ For more information on substituting graphic characters for alphabetic on a CBM, consult *PET/CBM Personal Computer Guide*, second edition, by Adam Osborne and Carroll S. Donahue (OSBORNE/McGraw-Hill, 1980).

Because it is sometimes hard to “see” a space character within a listing, we use an underlined period (it looks like this: .) to represent a space character which must be entered in an exact location or sequence; this usually occurs within a PRINT statement. Normally, we do not use a . character for a space which appears merely as a separator of words, but you should also enter these spaces for easier reading of your listings (unless the space is between words within curly braces, illustrated below). For now, you must remember to *enter a space character whenever you see the . character*. For example, the PRINT statement

```
PRINT "GREETINGS, EARTHLING! TAKE US TO YOUR LEADER!"
```

is shown as

```
PRINT "GREETINGS, .EARTHLING! .TAKE .US .TO .YOUR .LEADER!"
```

There are two kinds of special characters within curly braces: graphics characters (shift-whatever), and cursor control characters (for instance, cursor down, or home cursor). Graphics characters are shown within braces as their unshifted counterparts. Thus, the solid square would be shown as “{&}”, the diamond would be “{Z}”, and a shifted space would be shown as “{.}”.

Cursor control characters are shown within braces as the name of the key. The name of any cursor control key is printed in lower-case letters to avoid confusing the name with an odd sequence of graphics characters.

{up}	Cursor Up
{down}	Cursor Down
{left}	Cursor Left
{right}	Cursor Right
{rvs}	Reverse Video
{off}	Normal Video (reverse off)
{home}	Home Cursor
{clr}	Clear Screen
{ins}	Insert Character
{del}	Delete Character

Quite often, cursor control keys are repeated many times. For example, it's common to program a sequence like “{home down down down down down down}”. To save space in the printed listings, and to simplify your work when typing the programs, we tell you how many times to repeat a cursor control key. The “press-count” comes right before the key name, separated by a degree (°) character. Printed that way, the above example is listed out as “{home 7°down}”.

There is one catch to the “press-count” notation. There are a few graphics that get printed

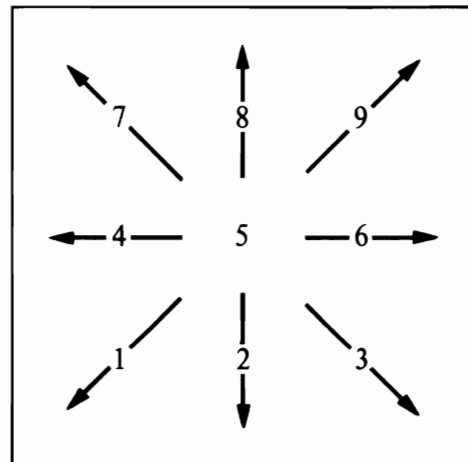
as numbers within braces. How can you tell when a digit inside braces represents a graphics character, and when it means a “press-count”? Since those particular graphics are not used much, a number within braces will most likely be a “press-count”, but the sure way to tell is to see if it is followed by a “°” and a cursor control. If so, it is a press-count. If it is followed by *anything* but a “°”, it is a graphics character. For example, when you see this line in a listing:

```
20 PRINT "{02. 3°left down}"
```

you enter: [2] [0] [space] [P] [R] [I] [N] [T]
[space] ['] ['] [shift-0] [shift-2] [shift-period]
[cursor left] [cursor left] [cursor left] [cursor
down] ['] ['].

Numeric Keypad

The numeric keypad is used to control movement in many of the games. As shown in the diagram below, you control movement on the screen by pressing the numeric key that has that position on the pad.



For example, to move an object up (or “forward”), press [8]. To move right, press [6], while [4] moves you left. To move down, press [2]. Some games also allow diagonal movement, using the [1], [3], [7] and [9] keys.

Sound

Several of the games use the CB2 sound capability of the PET. These games are much more fun if you have sound, so it is strongly suggested that you purchase a sound attachment. (If you have one of the new “Fat Forty” models, a sound device is built-in.) If your local

Commodore dealer doesn't have a sound adapter, write to Channel Data, 4141 State, Suite A2, Santa Barbara, CA, or to CAP Electronics, 8462 Hillwood Lane, Tucson, AZ 85715.

Another note about sound is that "CB2" is produced by using the 6522 chip, which also produces the tones that are used to record programs on the cassette recorder. If you use the Stop key to get out of a program that uses sound, you may find that you can't save a program. You can correct this program by typing the following line, which contains three commands which

restore the 6522 sound registers to their normal condition:

POKE 59464,0: POKE 59466,0: POKE 59467,0

Games that use sound are denoted in each program heading by the following symbol:





ACTION GAMES



ZAP

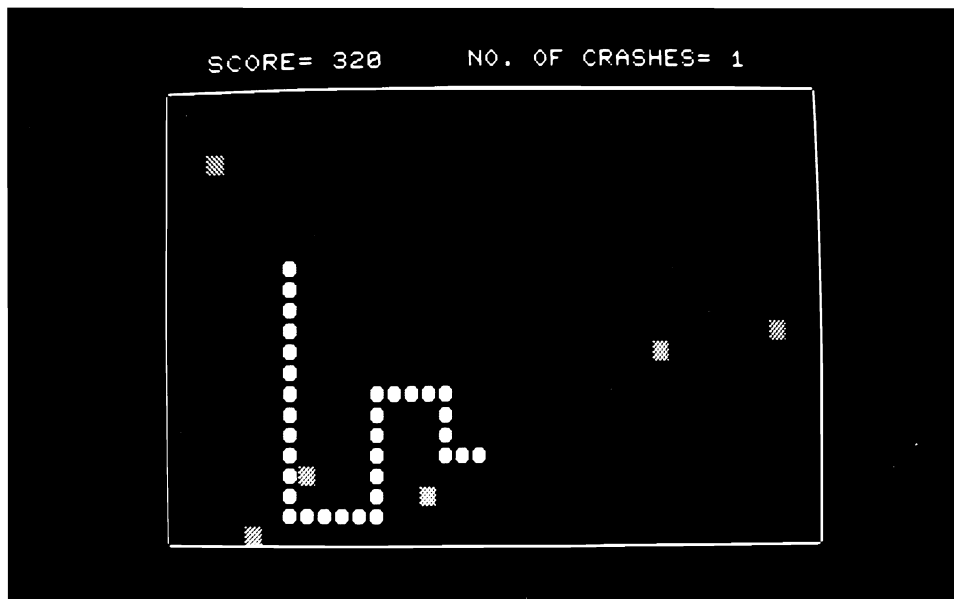
By Tom Marazita

When the game begins, a snake-like train of dots, (your zapper) starts crawling up the right-hand edge of the arena. Using your zapper, try to "zap" all the targets that pop up on your screen. The hitch is that you must do this without driving your "zapper" either into a wall or back across itself. After you get used to controlling your zapper, it will not be hard to avoid hitting a wall. However, each time you zap a target, your zapper becomes longer, and it becomes increasingly difficult to maneuver without running into yourself.

You control the zapper using the numeric

keypad. Each time you run over a target you get ten points. Pressing [5] allows you to stop where you are, but while you are stopped, your score automatically decreases very rapidly. If you spend a lot of time wandering about the arena but not zapping much of anything, many targets will appear and then disappear unzapped, thus reducing your final percentage score.

The game is over after you have crashed six times. The PET will tell you what your score was, what the maximum score for that session would have been if you had zapped all of the targets, and what percentage of the targets you actually did zap.



```
0 CLR:POKE 59468,12
6 REM ZAP COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81 RJ
90 PG$="ZAP":NM$="2":GOSUB 62000
100 DIM WM(255),A(255)
101 GOSUB 60400
102 EL=18:HP=EL-1
120 PRINT "{clr}"
130 PO=CRT+WD*25-2
140 S=1
160 FOR X=CRT+WD+1 TO CRT+WD*2-2:POKE X,100:NEXT
170 FOR X=CRT+WD*2 TO CRT+(WD*23) STEP WD:POKE X,103:NEXT
180 FOR X=CRT+(WD*24)+1 TO CRT+WD*25-2:POKE X,99:NEXT
```

```

190 FOR X=CRT+WD*24-1 TO CRT+WD*3-1 STEP -WD:POKE X,101:NEXT
195 GOSUB 900
200 GET A$:IF A$<>" " THEN 200
211 GOTO 410
220 A$="8":GOTO 240
230 GET A$:IF A$="Q" THEN 650
240 IF A$="4" THEN DIR=-1
250 IF A$="8" THEN DIR=-WD
260 IF A$="6" THEN DIR=1
270 IF A$="2" THEN DIR=WD
280 IF A$="7" THEN DIR=-WD-1
290 IF A$="9" THEN DIR=-WD+1
300 IF A$="3" THEN DIR=WD+1
310 IF A$="1" THEN DIR=WD-1
311 IF A$="5" THEN 770
320 PO=PO+DIR
321 IF PO=33767 OR PO=33728 OR PO=32808 OR PO=32847 THEN 590
324 IF PEEK(PO)=102 THEN GOSUB 550
325 WM(HP)=PO:HP=HP+1
326 IF HP>EL THEN HP=0
340 IF PEEK(PO)>98 AND PEEK(PO)<102 THEN 590
345 IF PEEK(PO)>102 AND PEEK(PO)<104 THEN 590
350 IF PEEK(PO)=81 THEN 590
360 POKE PO,81
370 R=INT(RND(1)*15)+1:IF S>5 THEN ZZ=20
375 P1=INT(RND(1)*ZZ)+1:ZZ=50
380 IF R=3 THEN 410
390 IF P1=3 THEN 480
400 GOSUB 730:GOTO 230
410 REM
420 IF S>6 THEN 400
430 Q=INT(RND(1)*1000)+(32768+160)
435 IF Q>33726 THEN 430
440 IF PEEK(Q)<>32 THEN 430
450 S=S+1:H=H+1:TC=TC+1
460 POKE Q,102
470 A(H)=Q:IF CV=1 THEN 400
475 CV=1:GOTO 220
480 REM
490 P=INT(RND(1)*H)+1
500 REM
510 IF PEEK(A(P))<>102 THEN 540
520 POKE A(P),32
530 S=S-1
540 A(P)=A(H):H=H-1:GOTO 400
550 REM
560 SC=SC+10
565 S=S-1
566 EL=EL+2
567 IF EL>250 THEN EL=250
570 GOSUB 900

```

```

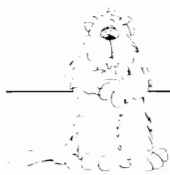
580 RETURN
590 CA=CA+1
595 GOSUB 900
600 IF CS>L THEN L=CS
610 CS=0
615 EL=18:HP=17
616 TP=0
620 CV=0
635 FOR T=1 TO 8:FOR X=1 TO 100:NEXT:POKE PO,209:FOR X1=1 TO 100:
    NEXT:POKE PO,81:NEXT
636 FOR X=0 TO 255:WM(X)=0:NEXT:HP=17:TP=0
638 IF CA=6 THEN 650
640 GOTO 120
650 PRINT "{clr 6°down}"
660 PRINT "YOUR..SCORE..IS"SC
662 PRINT
666 PRINT
667 PRINT "HIGHEST..POSSIBLE..SCORE:"TC*10
670 PRINT
675 PRINT "YOU..HIT";INT((SC/(TC*10))*100);"%..OF..THE..TARGETS"
677 GET Q$:IF Q$<>" " THEN 677:REM WAS POKE525,0
680 PRINT "{down}WANT..TO..PLAY..AGAIN?"
681 GET RJ$:IF RJ$="" THEN 681
682 IF RJ$<>"N" THEN CLR:GOTO 100
685 PRINT "{down}HOPE..YOU..HAD..FUN!{down}"
700 END
730 LOC=WM(TP):TP=TP+1
740 IF TP>EL THEN TP=0
750 POKE LOC,32
760 RETURN
770 FOR E=1 TO 171:GET A$:IF A$<>" " THEN 240
780 NEXT E:SC=SC-10:IF SC<0 THEN SC=0
790 GOSUB 900
800 GOTO 770
900 PRINT "{home 3°right}SCORE="SC;TAB(18);"NO..OF..CRASHES="CA
910 RETURN
60000 IN$="..":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
    GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
    Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$,:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
    GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
    NEXT Z:GOTO 60000

```

```

60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 GOSUB 60500:PRINT "{down}PRESS_{rvs}RETURN{off}_{TO}";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTIUNE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN CLR:PRINT "{clr}";:GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_{-}_CURSOR_{#}";NM$;
      "_{(C)}_{1981}_{PAGE}";PN;"{down}"
60270 GOSUB 60500:RETURN
60400 CRT=32768:CR$=CHR$(13):WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60420 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;"_{BY}_TOM_{MARAZITA}"
62010 PRINT "{down}_{CURSOR}_{#2}_{COPYRIGHT}_{(C)}_{1981}"
62020 GOSUB 60500
62025 PRINT "{down}CHASE_{THE}_{TARGETS}"
62030 PRINT "{2°down}WANT_{DIRECTIONS?}_{}";:GOSUB 60000
62040 IF IN$<>"Y" THEN 100
62100 LP=1:GOSUB 60200
62130 PRINT "YOU_{CONTROL}_{A}_{STRANGE}_{MISSILE}!"
62140 PRINT "{down}TARGETS_{'{'&}'_{APPEAR}_{ON}_{THE}_{SCREEN}_{,}_{WHICH}"
62150 PRINT "YOU_{TRY}_{TO}_{HIT}_{...}_{USE}_{NUMBERS}_{TO}_{CONTROL}"
62160 PRINT "THE_{MISSILE}_{:}_{8}_{MOVES}_{YOU}_{UP}_{,"
62170 PRINT "_{...}_{2}_{MOVES}_{YOU}_{DOWN}_{."
62180 PRINT
62190 PRINT "_{...}_{4}_{MOVES}_{YOU}_{LEFT}"
62200 PRINT "_{...}_{6}_{MOVES}_{YOU}_{RIGHT}_{."
62202 PRINT
62205 PRINT "(1,_{3},_{7},_{AND}_{9}_{MOVE}_{YOU}_{DIAGONALLY})"
62210 PRINT
62212 PRINT "EACH_{TARGET}_{IS}_{WORTH}_{10}_{POINTS}_{,}_{BUT}"
62225 PRINT "YOUR_{MISSILE}_{GETS}_{BIGGER}_{AFTER}_{HITTING}"
62228 PRINT "A_{TARGET}_{...}"
62230 PRINT "{down}THE_{GAME}_{ENDS}_{AFTER}_{6}_{CRASHES}!"
62235 GOSUB 60200

```



SHARK

By Glen Fisher

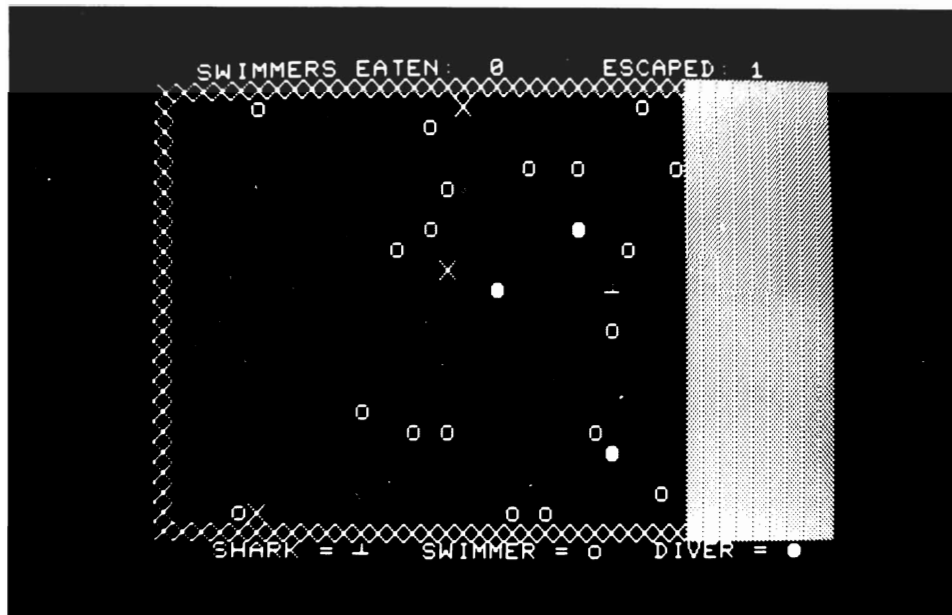
It's time for dinner, and we have just the thing — man-eating shark. Or is it shark, eating man? In this tasty little program, your task is to munch as many swimmers as you can. But beware of the divers and traps; they both make life difficult for you. Also, the swimmers are not at all cooperative — they're busy heading for the beach, trying to escape your jaws.

The PET will start by asking how many swimmers, how many traps, and how many divers you want. Make things as difficult or easy as you like (there are some limits). The swimming area is surrounded by shark traps on three sides and the beach on the fourth. Since "they" know you're there, "they've" sent out divers and put more shark traps into the swimming area, all in an effort to get you before you get them.

When your shark symbol ($\perp$) is flashing, it's

your turn to move (which is done using the numeric key pad). Being a shark, you're a pretty good swimmer, so you normally move two spaces. If you want to move only one space (to avoid a trap, perhaps), hold [SHIFT] down while you press the number key. After you move, each of the swimmers and divers gets to move one space.

To eat a swimmer, just attack him. Eating swimmers doesn't prevent you from moving — you may attack a swimmer next to you, and continue on to the space beyond (which may hold *another* swimmer). The hazards to avoid are traps, divers, and the beach itself. The divers are after you with spear guns. If they catch you, you're dead. If you attack a diver, you have a 50% chance of winning. Dinner is over when all the swimmers have either hit the beach or been eaten, or when you've been killed by a diver or a trap.



```
0 CLR:POKE 59468,12
6 REM SHARK (C) 1981 THE CODE WORKS
10 REM AS OF 10/06/81 RJ
90 PG$="SHARK":NM$="1":GOSUB 62000
100 GOSUB 60400:HT=25:SF=30:BT=(HT-2)*WD:RT=10
105 BL=32:SH=113:DV=81:SW=87:TR=86:BE=102
```



```

110 PRINT "{down}HOW.MANY.TRAPS?.";GOSUB 60000
115 IF IN$="" THEN IN$="5":PRINT "HOW.ABOUT.";IN$;".TRAPS, THEN?"
120 NT=VAL(IN$):IF NT>25 THEN
    PRINT "TRYING.TO.MAKE.SWIMMING.DANGEROUS?":GOTO 110
125 IF NT<1 THEN PRINT "THAT'S.A.CHALLENGE?":GOTO 110
130 PRINT "{down}HOW.MANY.SWIMMERS?.";GOSUB 60000:NS=VAL(IN$)
135 IF IN$="" THEN IN$="25":PRINT "OK, YOU'LL.GET.";IN$;".SWIMMERS."
137 NS=VAL(IN$)
140 IF NS>100 THEN PRINT "HEALTH.REGULATIONS.PROHIBIT.THAT.MANY.":
    GOTO 130
145 IF NS<1 THEN PRINT "ON.A.STARVATION.DIET?":GOTO 130
150 PRINT "{down}HOW.MANY.DIVERS?.";GOSUB 60000:ND=VAL(IN$)
155 IF IN$="" THEN IN$="2":PRINT "LET.THERE.BE.";IN$;".DIVERS."
160 ND=VAL(IN$):IF ND>25 THEN PRINT "DIVING.ISN'T.THAT.POPULAR!":
    GOTO 150
165 IF ND<1 THEN PRINT "SHARKS.AREN'T.AN.ENDANGERED.SPECIES":
    GOTO 150
185 ES=0:KD=0
186 DEF FN R(N)=INT((N-1)/3)-1:DEF FN C(N)=N-3*INT((N-1)/3)-2
187 DIM SS(NS,1),DS(ND,1)
188 FOR I=0 TO 500:NEXT I
189 PRINT "{clr}...SWIMMERS.EATEN:.....ESCAPED:"
190 FOR I=0 TO SF
191 POKE CRT+WD+I,TR:POKE CRT+I+BT,TR
192 NEXT I
193 FOR I=WD TO BT STEP WD
194 POKE CRT+I,TR
195 FOR J=SF+1 TO 39
196 POKE CRT+I+J,BE
197 NEXT J
198 NEXT I
199 FOR I=1 TO NT
200 GOSUB 10000
210 POKE LOC,TR
220 NEXT I
230 FOR I=1 TO NS
240 GOSUB 10000
245 SS(I,0)=ROW:SS(I,1)=COL
250 POKE LOC,SW
260 NEXT I
270 FOR I=1 TO ND
280 GOSUB 10000
285 DS(I,0)=ROW:DS(I,1)=COL
290 POKE LOC,DV
300 NEXT I
310 GOSUB 10000:POKE LOC,SH
320 SR=ROW:SC=COL
325 PRINT "{home
    24°down}...SHARK.=.{1}...SWIMMER.=.{W}...DIVER.=.{Q}";
330 PRINT "{home}":PRINT "{up}";TAB(18);KD;TAB(34);ES
500 GOSUB 10100

```

```

520 POKE OLD,BL
530 FOR I=1 TO DIST
540 OLD=OLD+CC+WD*RC
545 SC=SC+CC:SR=SR+RC
550 IF PEEK(OLD)=SW THEN GOSUB 10300:POKE OLD,BL:GOTO 570
560 IF PEEK(OLD)<>BL THEN GOTO 5000
570 NEXT I
580 POKE OLD,SH
590 IF ND<1 THEN 790
600 FOR I=1 TO ND
610 RM=HT+1:CM=WD+1
620 ROW=DS(I,0):COL=DS(I,1)
625 OLD=CRT+WD*ROW+COL
626 POKE OLD,BL
630 RC=SGN(SR-ROW):IF RC=0 THEN RC=1
631 CC=SGN(SC-COL):IF CC=0 THEN CC=1
635 LOC=OLD+RC*WD+CC
640 FOR J=RC TO -RC STEP -RC
650 FOR K=CC TO -CC STEP -CC
660 CH=PEEK(LOC):IF CH=BL OR CH=SH THEN 720
680 LOC=LOC-CC
690 NEXT K
700 LOC=LOC+3*CC-WD*RC
710 NEXT J
715 BUG=1:GOTO 50000
720 ROW=ROW+J:COL=COL+K
730 LOC=CRT+WD*ROW+COL
740 DS(I,0)=ROW:DS(I,1)=COL
750 CH=PEEK(LOC):POKE OLD,BL:POKE LOC,DV
760 IF CH=SH THEN CAUSE=1:GOTO 5100
765 IF CH<>BL THEN BUG=2:GOTO 50000
770 NEXT I
790 CT=NS
800 FOR I=1 TO NS
810 ROW=SS(I,0):COL=SS(I,1)
820 RC=SGN(ROW-SR):IF RC=0 THEN RC=SGN((HT-ROW)-ROW):IF RC=0 THEN
    RC=1
830 OLD=CRT+WD*ROW+COL
840 LOC=OLD+WD*RC+1
850 FOR J=RC TO -RC STEP -RC
860 FOR K=1 TO -1 STEP -1
865 CH=PEEK(LOC)
870 IF ABS(ROW+J-SR)+ABS(COL+K-SC)>1 AND (CH=BL OR CH=BE) THEN 930
880 LOC=LOC-1
890 NEXT K
900 LOC=LOC+3-WD*RC
910 NEXT J
920 J=0:K=0:LOC=OLD
930 ROW=ROW+J:COL=COL+K
935 POKE OLD,BL
940 IF PEEK(LOC)=BE THEN GOSUB 10500:GOTO 960

```

```

945 POKE LOC,SW
950 SS(I,0)=ROW:SS(I,1)=COL
960 NEXT I
970 IF CT=NS THEN 500
980 FOR I=1 TO CT
990 IF SS(NS,0)<0 THEN NS=NS-1:GOTO 990
1000 IF SS(I,0)<0 THEN SS(I,0)=SS(NS,0):SS(I,1)=SS(NS,1):NS=NS-1
1010 NEXT I
1020 GOTO 500
5000 CH=PEEK(OLD)
5010 IF CH=DV AND RND(1)<.5 THEN GOSUB 10300:GOTO 570
5030 IF CH=DV THEN CAUSE=1:GOTO 5100
5040 IF CH=BE THEN POKE OLD,SH+128:CAUSE=2:GOTO 5100
5050 IF CH=TR THEN CAUSE=3:GOTO 5100
5090 BUG=3:GOTO 50000
5100 FOR I=1 TO CAUSE:READ M$:NEXT I
5120 PRINT "{home 23°down}";
5130 PRINT ".....";CHR$(20);
    "{up}"
5140 PRINT TAB((30-LEN(M$))/2);"{rvs VVV off}";M$;"!!{rvs VVV
    off}"
5210 PRINT "DO YOU WANT TO PLAY AGAIN?.....{up}"
5215 PRINT TAB(27);:GOSUB 60000
5220 IF LEFT$(IN$,1)<>"N" THEN CLR:GOTO 100
5250 PRINT "{clr 6°down}THANKS FOR PLAYING."
5260 END
5500 DATA "THE DIVER GOT YOU"
5510 DATA "YOU BEACHED YOURSELF"
5520 DATA "YOU GOT TRAPPED"
5530 DATA "YOU GOT THE LAST SWIMMER"
5540 DATA "THE LAST SWIMMER JUST ESCAPED"
6010 END
10000 ROW=INT(RND(1)*(HT-3))+2:COL=INT(RND(1)*SF)+1
10010 LOC=CRT+WD*ROW+COL:IF PEEK(LOC)<>BL THEN 10000
10017 IF LOC>CRT+WD*25-1 THEN 10000
10020 RETURN
10100 OLD=CRT+WD*SR+SC:TM=TI:DK=-1
10105 GET C$:IF C$<>" " THEN 10105
10110 GET C$:IF C$<>" " THEN 10160
10120 IF TI<TM THEN 10110
10130 IF DK THEN POKE OLD,BL
10140 IF NOT DK THEN POKE OLD,SH
10150 DK=NOT DK:TM=RT+TI:GOTO 10110
10160 TE=(ASC(C$) AND 127)-ASC("0")
10165 IF C$="Q" THEN 5250
10170 IF TE<1 OR TE>9 THEN 10110
10180 DIST=1:RC=FN R(10-TE):CC=FN C(TE)
10190 IF ASC(C$)<128 THEN DIST=2
10200 RETURN
10250 TM=TI+DL
10260 IF TI<TM THEN 10260

```

```

10270 RETURN
10300 CH=PEEK(OLD):DL=5
10305 POKE OLD,SH
10310 FOR II=1 TO 4
10320 GOSUB 10250:POKE OLD,CH
10330 GOSUB 10250:POKE OLD,SH
10340 NEXT II
10350 POKE OLD,BL
10360 IF CH=SW THEN 10420
10370 FOR II=1 TO ND
10380 IF DS(II,0)=SR AND DS(II,1)=SC THEN 10400
10390 NEXT II:BUG=4:GOTO 50000
10400 DS(II,0)=DS(ND,0):DS(II,1)=DS(ND,1):ND=ND-1
10410 RETURN
10420 FOR II=1 TO NS
10430 IF SS(II,0)=SR AND SS(II,1)=SC THEN 10450
10440 NEXT II:BUG=5:GOTO 50000
10450 SS(II,0)=SS(NS,0):SS(II,1)=SS(NS,1):NS=NS-1:KD=KD+1
10455 PRINT "{home}";TAB(18);STR$(KD);".."
10460 IF NS>0 THEN RETURN
10470 POKE OLD,SH:CAUSE=4:GOTO 5100
10500 DL=1:LOC=CRT+WD*ROW+(SF+1)
10510 FOR II=SF+1 TO 39
10520 POKE LOC,SW+128:GOSUB 10250
10530 POKE LOC,BE:LOC=LOC+1
10540 NEXT II
10550 SS(I,0)=-1
10560 CT=CT-1:ES=ES+1
10565 PRINT "{home}";TAB(34);STR$(ES);".."
10570 IF CT>0 THEN RETURN
10580 CAUSE=5:GOTO 5100
20000 DL=20
20010 GOSUB 10250:POKE LOC,CH
20020 GOSUB 10250:POKE LOC,DV
20030 GOTO 20010
20100 DL=16:CH=PEEK(OLD)
20110 GOSUB 10250:POKE OLD,(CH+128) AND 255
20120 GOSUB 10250:POKE OLD,CH
20130 GOTO 20110
50000 T$=STR$(BUG):T=LEN(T$)+3
50010 PRINT "{home , off}";
50020 FOR IJ=1 TO T:PRINT "{"";:NEXT IJ
50030 PRINT "{,}":PRINT "{rvs !}BUG";T$;"{off !}"
50040 PRINT "{< rvs}";
50050 FOR IJ=1 TO T:PRINT "{"";:NEXT IJ
50060 PRINT "{off >}"
60000 IN$="..":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010

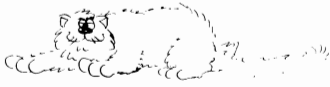
```

```

60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>10 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 GOSUB 60500:PRINT "{down}PRESS_{rvs}RETURN{off}_{TO}";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN CLR:PRINT "{clr}";:GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_{CURSOR}_{#}";NM$;
      "_{(C)}_{1981}_{PAGE}";PN;"{down}"
60270 GOSUB 60500:RETURN
60400 CRT=32768:CR$=CHR$(13):PRINT "{clr}"
60410 WD=40:POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:
      PRINT CHR$(142)
60420 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT "{left ins D}":RETURN
62000 PRINT "{clr 2°down}";TAB(9);PG$;"_{BY}_{GLEN}_{FISHER}"
62010 PRINT "{down}_{CURSOR}_{#1}_{COPYRIGHT}_{(C)}_{1981}"
62020 GOSUB 60500
62030 PRINT "{down}EAT_{ALL}_{THE}_{SWIMMERS}_{BEFORE}_{THEY}_{ESCAE}{left
      ins}P"
62040 PRINT "{2°down}WANT_{DIRECTIONS?}_{}";:GOSUB 60000
62050 IF IN$<>"Y" THEN 100
62100 LP=2:GOSUB 60200
62130 PRINT "YOU_{ARE}_{A}_{SHARK}_{'1}'"
62140 PRINT "THAT_{EATS}_{SWIMMERS}_{'W}'"
62150 PRINT "WHILE_{AVOIDING}_{DIVERS}_{'Q}'"
62160 PRINT "AS_{WELL}_{AS}_{TRAPS}_{'V}'"
62170 PRINT
62180 PRINT "USE_{THE}_{NUMBER}_{KEYS}_{TO}_{MOVE}."
62185 PRINT ""
62190 PRINT "THE_{'5'}_{KEY}_{IS}_{YOUR}_{CURRENT}_{LOCATION}."
62195 GOSUB 60200
62200 PRINT "FOR_{EXAMPLE}_{,}_{TO}_{MOVE}_{TWO}_{SPACES}"
62210 PRINT "TO_{THE}_{RIGHT}_{,}_{PRESS}_{THE}_{'6'}_{KEY}"
62220 PRINT
62230 PRINT "_{7}_{8}_{9}"
62240 PRINT "_{M}_{B}_{N}"
62250 PRINT ""
62260 PRINT "_{4}{C}_{5}_{C}6"
62270 PRINT ""

```

```
62280 PRINT ".....{N}..{B}..{M}"
62290 PRINT ".....1..2..3"
62300 PRINT
62310 PRINT "SHIFTING..A..NUMBER..MOVES..ONE..SPACE"
62320 PRINT "IN..THAT..DIRECTION..(INSTEAD..OF..TWO)"
62325 GOSUB 60200
```



DEMON

By Ken Morley



Your task is to recapture a demon that has accidentally escaped during the demonstration of a spell. As long as the demon is avoiding capture, he can't free any more demons. The problem is that, as long as the spell is active, another demon can escape once

the first one is recaptured. Thus, you've got to keep after demons until the spell wears off. When the spell expires, the first escaped demon will be sent back where he belongs, but any demons he has freed will remain. It is, therefore, vital that he not be given the chance to free any other demons.

```
0 CLR:POKE 59468,12
6 REM DEMON! COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81 RJ
90 PG$="DEMON!":NM$="11":GOSUB 62000
100 REM
110 FOR I=1 TO 9:READ N2(I):NEXT I
120 DATA 17,14,13,8,0,12,9,10,11
130 FOR I=1 TO 8:D(I)=I-(I>4):NEXT I
140 T=WD:FOR I=2 TO 9 STEP 3:FOR J=-1 TO 1:E(I+J)=T+J:NEXT:T=T-WD:
    NEXT I
160 FOR I=1 TO 9:READ N1(I):NEXT
170 DATA 58,52,46,28,0,43,30,34,38
190 FQ=59464:TN=59466:CT=59467
300 NM=0:CR$=CHR$(13)
310 OI=33708:TI$="000000"
320 PRINT "{clr}"
480 GOTO 500
490 PRINT "{down}PLEASE ANSWER '1','2',OR '3'."
500 PRINT "{down}WANT A (1) SHORT, (2) MEDIUM,OR"
510 PRINT "(3) LONG GAME?";GOSUB 60000
520 IF IN$="" THEN PRINT "I'LL ASSUME A SHORT GAME.":FOR T=1 TO
    1000:NEXT:IN$="1"
530 IF LEN(IN$)>1 THEN 490
540 IF IN$<"1" OR IN$>"3" THEN 490
550 LM=3600*VAL(IN$)
800 GOSUB 10000
810 FOR I=1 TO 5:POKE OI,81:FOR J=1 TO 100:NEXT:POKE OI,32:
    FOR J=1 TO 100:NEXT:NEXT I
820 POKE OI,81
900 OP=32809+INT(RND(1)*918)
910 IF PEEK(OP)<>32 THEN 900
1000 ND=8
1005 T=INT(RND(1)*ND):DN=D(T):D(T)=D(ND):D(ND)=DN:ND=ND-1
1006 IF PEEK(OP+E(DN))<>32 THEN 1005
1010 FOR I=1 TO INT(RND(1)*6)+3
1020 NP=OP+E(DN)
```

```

1030 IF PEEK(NP)=81 THEN GOSUB 5000:GOTO 900
1040 IF PEEK(NP)=32 THEN POKE OP,32:POKE NP,87:OP=NP:PC=N2(DN):
      GOSUB 4000
1100 IF TI>LM THEN GOTO 2000
1105 POKE QP,255:GET B$:IF B$="." THEN 2000
1110 B=5:IF B$<>" " THEN B=VAL(B$):IF B=0 THEN B=5
1120 N1=O1+E(B)
1130 IF PEEK(N1)=87 THEN GOSUB 5000:GOTO 900
1140 IF PEEK(N1)=32 THEN POKE O1,32:POKE N1,81:O1=N1:PC=N1(B):
      GOSUB 4000
1500 NEXT I
1510 GOTO 1000
2000 PC=0:GOSUB 4000:POKE TN,1:PC=255
2001 POKE OP,102:GOSUB 2500
2002 POKE OP,224:GOSUB 2500
2003 POKE OP,32:GOSUB 2500
2004 GOSUB 4100
2010 FOR I=1 TO 1000:NEXT I
2015 PRINT "{clr 3°down}THE SPELL HAS EXPIRED!"
2020 PRINT "{down}YOU CAUGHT";NM;"DEMON";
2025 IF NM<>1 THEN PRINT "S";
2030 PRINT "!":FOR I=1 TO 100:NEXT I
2035 GET T$:IF T$<>" " THEN 2035
2040 PRINT "{down}WANT TO PLAY AGAIN?.";:GOSUB 60000
2050 IF LEFT$(IN$,1)<>"N" THEN 300
2100 PRINT "{clr 2°down}":END
2500 FOR I=1 TO 20:FOR J=1 TO 10:NEXT J:POKE FQ,PC:PC=PC-4:NEXT I:
      RETURN
4000 POKE CT,16:POKE TN,15:POKE FQ,PC:IF PC=0 THEN RETURN
4100 POKE FQ,0:POKE TN,0:POKE CT,0:RETURN
5000 POKE OP,215
5010 PC=0:GOSUB 4000
5020 FOR K=1 TO 250 STEP 2:POKE FQ,K:NEXT
5030 GOSUB 4100
5040 NM=NM+1
5050 RETURN
10000 T$="":FOR I=1 TO WD-3:T$=T$+"{@}":NEXT I
10010 PRINT "{clr 0}";T$;"{. left ins @}"
10020 FOR I=1 TO 23:PRINT "{|}";TAB(WD-2);"{| left ins}":NEXT I
10030 PRINT "{-}";T$;"{= left ins @ home}"
10035 FOR K=1 TO 70
10040 W=INT(RND(1)*(WD*23-2)+WD+1)
10050 IF PEEK(32768+W)<>32 THEN 10040
10060 POKE 32768+W,214
10070 NEXT
10080 RETURN
60000 IN$="." :ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010

```



```

60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{to}";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;"{_}CURSOR_{#}";NM$;
      "{_}(C)_{1981}_{_}PAGE";PN
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}";:GOSUB 60400:GOTO 100
60400 CRT=32768:WD=40:CR$=CHR$(13):PRINT "{clr}":QP=515:
      IF PEEK(50000) THEN QP=151
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60420 RETURN
60500 FOR I=1 TO 13:PRINT "@@@";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;"{_}BY_{KEN}_{MORLEY}"
62005 PRINT "{down}_{_}_{_}_{_}_{_}CURSOR_{#11}_{_}COPYRIGHT_{_}(C)_{1981}"
62010 GOSUB 60500
62030 PRINT "{down}CAPTURE_{THE}_{ESCAPED}_{DEMONS}"
62040 PRINT "{2°down}WANT_{DIRECTIONS?}_{_}";:GOSUB 60000
62050 IF LEFT$(IN$,1)<>"Y" THEN 60300
62100 LP=3:GOSUB 60200
62110 PRINT "IN_{THE}_{MIDST}_{OF}_{A}_{DEMONSTRATION}_{OF}_{A}_{NW{left ins}E}"
62120 PRINT "{down}SPELL_{AT}_{THE}_{WIZARDS}_{WARLOCKS}_{AND}_{SOR-"
62130 PRINT "{down}CERERS_{CHOWDER}_{AND}_{MARCHING}_{SOCIETY}_{AN}"
62140 PRINT "{down}ACCIDENT_{OCCURRED}_{PERMITTING}_{THE}_{ESCAE{left
      ins}P"
62150 PRINT "{down}OF_{SOME}_{DEMONS}_{_}FORTUNATELY,{_}THE_{SPELL}"
62155 PRINT "{down}WAS_{SET}_{UP}_{SO}_{THAT}_{IT}_{WILL}_{EXPIRE}_{SOON}."
62160 PRINT "{down}ALSO,{_}ONLY_{ONE}_{DEMON}_{CAN}_{BE}_{LOOSE}_{AT}_{A}"
62170 PRINT "{down}TIME_{(WHICH}_{IS}_{JUST}_{AS}_{WELL}_{_}_{_}YOUR"
62180 PRINT "{down}TASK_{IS}_{TO}_{RECAPTURE}_{AS}_{MANY}_{DEMONS}_{AS}"
62190 PRINT "{down}POSSIBLE_{BEFORE}_{THE}_{SPELL}_{EXPIRES}."
62200 GOSUB 60200
62201 PRINT "THE_{DEMONS}_{MUST{4°left down ###
      up}_{_}BE_{RECAPTURED}_{IF}_{LEFT}"
62202 PRINT "{down}FREE,{_}THEY_{WILL}_{TRY}_{TO}_{FREE}_{STILL}_{MORE}"

```

```

62203 PRINT "{down}DEMONS..THAT.CANNOT{6°left down #####
        up}.BE.ALLOWED!"
62210 PRINT "{2°down}YOU.MAY.CONTROL.YOUR.MOVES.WITH.THE"
62220 PRINT "{down}NUMERIC.PAD..THE.NUMBER.PAD.....7.8.9"
62230 PRINT TAB(35);"{M|N}"
62240 PRINT "MOVES.YOU.AS.SHOWN.TO.THE.RIGHT...4{@Q@}6"
62250 PRINT TAB(35);"{N|M}"
62260 PRINT "TO.STOP.THE.GAME.EARLY,.PRESS.....1.2.3"
62270 PRINT "{down}THE.{rvs}SPACE{off}.KEY."
62310 GOSUB 60200
62320 PRINT "IN.THE.GAME,.YOU.WILL.BE.THE.'{Q}',.AND"
62330 PRINT "{down}THE.CURRENTLY.ESCAPED.DEMON.WILL.BE.'{W}'.{left
        ins}'"
62340 PRINT "{down}WHEN.YOU.CAPTURE.A.DEMON,.IT.IS.FROZEN"
62350 PRINT "{down}IN.PLACE,.AND.MARKED.WITH.'{rvs W off}'.WHEN.THE"
62360 PRINT "{down}SPELL.ENDS,.THE.CURRENTLY.ESCAPED.DEMON"
62370 PRINT "{down}WILL.BE.SENT.BACK.WHERE.HE.CAME.FROM,"
62380 PRINT "{down}AND.WON'T.BE.A.PROBLEM..ANY.DEMONS.WHO"
62390 PRINT "{down}WERE.FREED.BY.A.DEMON.WILL.REMAIN,.AND"
62400 PRINT "{down}WILL.BE.A.VERY.LARGE{5°left down #####
        up}.PROBLEM.{down}"
62410 GOSUB 60200

```



FERRY



By John Matarella

Your mission is to ferry supplies through an asteroid belt to a starving, desperate mining base in space. Since the people at the base are dying by the minute, you must try to cross through the belt as fast as you can. You have five remote-controlled ships that travel one at a time across the asteroid belt.

Your ships can travel forward and backward as you try to avoid asteroids. You move forward by

pressing [6], and backward by pressing [4]. When neither key is pressed, the craft doesn't move, but an encounter with an asteroid means the end of that ship. Since the asteroids are always moving, you must be quick to survive.

Your score is based on your speed, how far you get across the asteroid belt, and how many ships survive. Your high score is saved, so you can play several games and watch yourself improve.

```
0 CLR:POKE 59468,12
6 REM FERRY COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81 RJ
90 PG$="FERRY!":NM$="13":GOSUB 62000
100 CL=59468:MS=2
110 DEF FN R(X)=INT(RND(1)*X+1)
120 R=5:NS=5
130 SD=0:PRINT "{2°down}WANT SOUND?_";:GOSUB 60000
140 IF IN$="" THEN PRINT "{up 12°right}N"
150 IF LEFT$(IN$,1)="Y" THEN SD=16
310 PRINT "{down}HOW HARD_(1=EASY,_9=HARD)_?_";:GOSUB 60000
320 IF IN$="" THEN IN$=STR$(R):PRINT "{up}"TAB(28)R:FOR I=1 TO 500:
NEXT
330 R=VAL(IN$):IF R<1 OR R>9 GOTO 310
335 SF=(R-1)/8*(MS-1)+1
340 TD=0:W=0:YY=0:L=0:T=0:TL=0
350 P=32768+R*WD*2
360 SS=P
370 PRINT "{clr}"
380 FOR I=1 TO 24:PRINT TAB(FN R(37)+1);".{up}":PRINT TAB(
FN R(37)+1);".":NEXT I
400 T=T+1:CC=0
405 POKE 59467,SD:POKE 59464,0:POKE 59466,15
420 CC=CC+1
430 POKE P,62
440 OP=P
450 POKE QP,255:GET A$:A=0:IF A$<>" " THEN A=ASC(A$):IF A$="Q" THEN
POKE 59467,0:GOTO 800
455 XY=15:XX=255:IF A=52 OR A=54 THEN XX=128+SS-P:XY=9:IF A=52 THEN
XY=5
460 POKE 59464,XX:POKE 59466,XY
465 IF A=54 THEN P=P+1
466 IF A=52 THEN P=P-1
480 IF P>=SS+39 THEN 710
```

```

490 IF P<SS THEN P=SS
500 POKE OP,32
520 PRINT TAB(FN R(37)+1);".{up}":PRINT TAB(FN R(37)+1);"."
530 IF PEEK(P)<>46 THEN 420
570 L(T)=P-SS
575 PRINT "{home}";TAB(25);"YOU_BLEW_IT!":PRINT TAB(25);"SHIP_#";T
580 POKE CL,12
590 POKE 59467,SD:POKE 59466,15
600 FOR I=50 TO 250 STEP 3
610 POKE P,42
620 POKE 59464,I
630 POKE P,62
640 NEXT I
670 GOTO 730
710 POKE OP,32:XX=0
715 PRINT "{home}";TAB(25);"SUCCESS!":PRINT TAB(25);"SHIP_#";T
720 POKE QL,24
725 W=W+1:L(T)=40
730 POKE 59466,0:POKE 59467,0
740 POKE P,32
745 GET T$:IF T$<>" " THEN 745
750 D=((P-SS)↑2)/CC:TD=TD+D:GOSUB 1300:PRINT "{home}SCORE:";YY
755 D=TD:GOSUB 1300:PRINT "TOTAL_SO_FAR:";YY
760 PRINT "{home 23°down}PRESS_ANY_KEY_TO_";
765 IF T=NS THEN PRINT "SEE_SUMMARY."
770 IF T<NS THEN PRINT "LAUNCH_NEXT_SHIP."
775 GET T$:IF T$="" THEN 775
777 IF T$="Q" THEN 800
780 IF T<NS THEN P=SS:GOTO 350
800 PRINT "{clr down}YOU_WERE_GIVEN";T;"SHIPS_TO_PILOT_TO"
805 PRINT "THE_ASTEROID_BASE_";W;"SHIP";IF W<>1 THEN PRINT "S";
808 PRINT "_SURVIVED."
810 D=TD:GOSUB 1300
815 IF W=0 AND SD>0 THEN GOSUB 900
830 PRINT "{2°down}YOUR_SCORE=";YY
835 IF YY>HY THEN HY=YY
837 PRINT "{2°down}HIGHEST_SCORE=_";HY
850 GET T$:IF T$<>" " THEN 850
860 PRINT
870 PRINT "DO_YOU_WANT_TO_PLAY_AGAIN?_";
875 GOSUB 60000
885 IF LEFT$(IN$,1)<>"N"GOTO 310
890 END
900 RESTORE:A=20
905 PRINT "{down}WE_SHALL_NOW_HONOR_THOSE_COMPUTERS_THAT"
906 PRINT "GAVE_THEIR_LIVES_FOR_THE_TRAINING_OF"
907 PRINT "THIS_(AHM...)_PILOT."
910 POKE 59467,16:POKE 59464,0:POKE 59466,104
920 READ C,T,Z
930 T=T*A:Z=Z*A
940 POKE 59464,C:FOR I=1 TO T:NEXT I

```

```

950 POKE 59464,0:FOR I=1 TO Z:NEXT I
960 IF Z>0 THEN 920
970 POKE 59466,0:POKE 59467,0:RETURN
1200 DATA 237,10,1,237,5,1,177,20,5,237,10,1,177,5,1,140,20,5
1210 DATA 237,10,1,177,5,1,140,15,2,237,10,1,177,5,1,140,15,2
1220 DATA 237,10,1,177,5,1,140,25,5,177,10,1,140,5,1,118,25,1
1230 DATA 140,20,1,177,15,1,237,25,1,237,20,1,237,10,1,177,25,0
1300 YY=INT((D*SF)/(39*MS*NS)*1000):RETURN
60000 IN$=" ":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$(" _{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT " _{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off} _TO_ ";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;" _- _CURSOR_#";NM$;
      " _ _ (C) _1981 _ _ _PAGE";PN
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 CRT=32768:CR$=CHR$(13):WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60420 QP=515:QL=245:CR$=CHR$(13)
60430 IF PEEK(50000) THEN QP=151:QL=216
60450 RETURN
60500 FOR I=1 TO 13:PRINT " {@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;" _ _BY _JOHN _MATARELLA"
62010 PRINT "{down} _ _ _ _CURSOR_#13 _ _COPYRIGHT _ (C) _1981"
62020 GOSUB 60500
62030 PRINT "{down}CARRY _SUPPLIES _TO _THE _ASTEROID _OUTPOST"
62080 PRINT "{2°down}WANT _DIRECTIONS? _";:GOSUB 60000
62090 IF LEFT$(IN$,1)<>"Y" THEN 60300
62100 LP=3:GOSUB 60200
62110 PRINT "FAR _OUT _IN _THE _ASTEROIDS, _A _SMALL _GROUP"

```

```

62120 PRINT "{down}OF PEOPLE HAVE SET UP A MINING COMPANY."
62130 PRINT "{down}LIKE MOST OTHER PEOPLE, THEY NEED TO ET{left
ins}A"
62140 PRINT "{down}OCCASIONALLY...YOU HAVE BEEN GIVEN THE"
62150 PRINT "{down}RESPONSIBILITY OF DELIVERING NEEDED"
62160 PRINT "{down}SUPPLIES TO THE BASE OUT IN THE ASTER-"
62170 PRINT "{down}OIDS."
62180 GOSUB 60200
62190 PRINT "SINCE PILOTING THROUGH THE ASTEROID BET{left ins}L"
62200 PRINT "{down}IS HAZARDOUS, AND THERE ARE TOO MANY"
62210 PRINT "{down}PEOPLE INVOLVED TO ALLOW ANY RISK TO"
62220 PRINT "{down}LIFE, REMOTE CONTROL SHIPS{up left ### 2°left
down} ARE USED."
62230 PRINT "{2°down}YOUR TASK IS TO NAVIGATE THE SUPPLY"
62240 PRINT "{down}SHIPS THROUGH THE ASTEROIDS TO THE BAS.{left
ins}E"
62250 PRINT "{down}THERE ARE FIVE SHIPS, AND THEY FLY ONE-"
62260 PRINT "{down}AT-A-TIME (THE EASIER TO CONTROL THEM)."
62270 GOSUB 60200
62280 PRINT "YOUR SCREEN WILL SHOW YOU THE PATH OF A"
62290 PRINT "{down}SHIP, AND THE ASTEROIDS THAT IT PASSES."
62300 PRINT "{down}TO MOVE THE SHIP FORWARD (RIGHT), PRESS"
62310 PRINT "{down}THE_{rvs}6{off left up $
down}_KEY, AND TO MOVE IT BACK (LEFT)"
62320 PRINT "{down}PRESS THE_{rvs}4{off left up $ down}_KEY."
62330 PRINT "{2°down}YOU WILL BE AWARDED A RATING DEPENDING"
62340 PRINT "{down}ON HOW QUICKLY YOU GET THE SUPPLIES"
62350 PRINT "{down}THROUGH, AND HOW FEW SHIPS YOU DESTROY."
62360 PRINT TAB(14); "{2°down}GOOD LUCK!"
62370 GOSUB 60200

```



BAT

By Chris Nadovich



Life isn't easy for a bat, as you'll find out in this game. You have to avoid bouncing off walls, make quick decisions about good-tasting and bad-tasting bugs, keep beating those little wings, and avoid starving to death. As a bat, you need to fly around and eat insects. Some of the insects are tasty, and will help you stay alive. But the so-called "dirt bug" is worthless junk food, and "lint" actually reduces your current calories by half.

The insects have the following values:

☒	Elephant bug	25 calories
♣	Buffet beetle	10 calories
♥	Feast fly	doubles calories
♦	Lint	halves calories
♠	Dirt bug	worthless

To liven things up a bit, the radar of this

species of bat doesn't work very well, so it has evolved a very effective ability to fly into walls in the dark and bounce off.

To control the bat, you use the numeric pad in the usual manner, so that [8] means up, [6] means right, and so on. Note that pressing the button merely applies force; it does not necessarily cause an immediate change in direction. For example, if the bat is moving left, and you hold down [6] to move right, it must first slow down and stop before it starts moving right. That's why this game is so difficult.

To apply a force for a long time, just hold down the button for a while. You don't need to let it up, except to change the direction of the applied force.

Remember, all the time you're flying, you're burning calories. We'll put 50 calories in your tank to start you off.

```

0 CLR:POKE 59468,12
6 REM BAT! COPYRIGHT(C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81 RJ
90 PG$="BAT!":NM$="14":GOSUB 62000
100 PRINT "{down}WANT SOUND?";:GOSUB 60000
101 IF IN$="" THEN PRINT TAB(12);"{up}N":FOR I=1 TO 500:NEXT
105 PRINT "{clr}"
110 SD=0:IF LEFT$(IN$,1)="Y" THEN SD=16
120 POKE 59467,SD:POKE 59466,15:POKE 59464,0
130 DIM T(29):FOR I=0 TO 29:T(I)=INT(257/(2^(I/12))-1.5):NEXT I
140 Q8=59464
160 D(1)=26:D(2)=18:D(3)=25:D(4)=42:D(6)=41:D(7)=58:D(8)=50:D(9)=57
200 G=9.8:REM GRAVITATIONAL ACCEL.
210 U=.3:REM FRICTION COEFF.
220 A=20:REM THRUST ACCEL.
250 U=1-U:G=G/(9.8*4):A=A/(9.8*4)
500 RESTORE:GF=.142:TP=50
510 TQ=TI:ZQ=59456:ZW=32:S0=8*16*16*16
600 PRINT "{clr &&&&&}CALORIES:.....{&&&&}TIME:.....{&&&&&}";
610 FOR I=1 TO 23:PRINT "{left down &}";:NEXT
620 FOR I=1 TO 39:PRINT "{2°left &}";:NEXT
630 FOR I=1 TO 24:PRINT "{up left &}";:NEXT
640 PRINT "{home}":GOSUB 2000
700 X=20:Y=12:VY=0:VX=0
1000 K=S0+INT(X+.5)+WD*INT(Y+.5)

```

```

1010 R=PEEK(K):TP=TP-GF*(TP/50+1)
1020 IF Q9=1 THEN N=255:Q9=0:GOTO 1040
1030 IF Q9=0 THEN N=127:Q9=1:GOTO 1040
1040 IF R=32 GOTO 1150
1090 IF R=214 THEN TP=TP+25:POKE Q8,T(21):GOTO 1140
1100 IF R=88 THEN TP=TP+10:POKE Q8,T(24):GOTO 1140
1110 IF R=83 THEN TP=TP*2:POKE Q8,T(29):GOTO 1140
1120 IF R=42 THEN POKE Q8,T(1):TP=TP/2
1140 GOSUB 4000
1150 POKE K,N
1180 POKE QP,255:GET Q$:IF Q$="" THEN 1600
1182 IF Q$="Q" THEN 3020
1185 C=ASC(Q$)-48
1190 TP=TP-GF*A
1250 IF C=6 THEN POKE Q8,T(9):VX=VX+1*A:Q=Q-128:GOTO 1600
1300 IF C=2 THEN POKE Q8,T(17):VY=VY+1*A:Q=Q-64:GOTO 1600
1400 IF C=4 THEN POKE Q8,T(12):VX=VX-1*A:Q=Q-32:GOTO 1600
1500 IF C=8 THEN POKE Q8,T(5):VY=VY-1*A
1600 VY=VY+G:NY=Y+VY:NX=X+VX:POKE Q8,0
1700 IF NX>38 THEN NX=38:VY=U*VY:VX=-VX*U
1750 IF NX<1 THEN NX=1:VY=U*VY:VX=-VX*U
1800 IF NY>22 THEN NY=22:VX=U*VX:VY=-VY*U
1850 IF NY<1 THEN NY=1:VX=U*VX:VY=-VY*U
1900 Y=NY:X=NX
1910 IF TP<0 GOTO 3000
1920 N=32:POKE K,N
1930 IF Q9=1 GOTO 1990
1940 POKE Q8,0
1950 PRINT "{home}";TAB(15);INT(TP);"{left}_":PRINT "{home}";
      TAB(28);INT((TI-TQ)/60);"{left}_":
1990 GOTO 1000
2000 FOR I=1 TO 5
2010 READ BG,BN
2015 GB(I)=BG
2020 FOR J=1 TO BN
2030 BX=2+INT(36*RND(1))
2040 BY=2+INT(21*RND(1))
2050 POKE (S0+BX+BY*WD),BG
2060 NEXT
2070 NEXT
2080 DATA 214,5,88,4,42,5,83,4,65,3
2999 RETURN
3000 REM DUST TO DUST
3001 PRINT "{home 10°down}";TAB(14);"{rvs}YOU ARE DEAD{off}"
3002 RESTORE
3003 FOR I=1 TO 5:READ Q5,Q4:NEXT
3005 FOR K=1 TO 11
3006 READ Q5,Q4:Q4=200*Q4
3007 POKE Q8,T(Q5)
3008 FOR J=1 TO Q4:NEXT
3009 POKE Q8,0

```



```

3010 NEXT
3012 DATA 5,4,5,3,5,1,5,4,8,3,7,1,7,3,5,1,5,3,4,1,5,4
3020 GET RJ$:IF RJ$<>" " THEN 3020
3030 PRINT "{clr 2°down}WANT_TO_TRY_AGAIN?";:GOSUB 60000
3040 IF LEFT$(IN$,1)<>"N" THEN 500
3050 POKE 59467,0:PRINT "{clr}":END
4000 BX=2+INT(36*RND(1))
4040 BY=2+INT(21*RND(1))
4050 POKE (S0+BX+BY*WD),GB(INT(RND(1)*5+1))
4999 RETURN
60000 IN$=" ":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{to}";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;"_{cursor}#";NM$;
      "._.(C).1981._.PAGE";PN
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 CRT=32768:CR$=CHR$(13):WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60420 QP=515:CR$=CHR$(13)
60430 IF PEEK(50000) THEN QP=151
60440 RETURN
60500 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;"_{by}.C.T._NADOVITCH"
62010 PRINT "{down}._.CURSOR.#14._.COPYRIGHT._.(C).1981"
62020 GOSUB 60500
62030 PRINT "{down}REDUCE_THE_LOCAL_INSECT_POPULATION"
62080 PRINT "{2°down}WANT_DIRECTIONS?";:GOSUB 60000
62090 IF LEFT$(IN$,1)<>"Y" THEN 60300

```

```

62100 LP=4:GOSUB 60200
62110 PRINT "OVER THE YEARS, RADIO-CONTROLLED MODELS{down}"
62120 PRINT "HAVE BECOME EVER MORE SOPHISTICATED. TE{left
ins}H{down}"
62130 PRINT "LATEST DEVELOPMENTS ARE SO LIFE LIKE. THT{left
ins}A{down}"
62140 PRINT "THEY SEEM ALIVE. YOU WILL SOON HAVE A{down}"
62150 PRINT "CHANCE TO TRY ONE OUT. THE MODEL, CALLD{left
ins}E{down}"
62160 PRINT "THE 'SUPERBAT', COMES IN THE FORM OF A{down}"
62170 PRINT "BAT, AND IS ALSO A GAME OF SKILL. THE{down}"
62180 PRINT "AIR IS FILLED WITH 'INSECTS' FROM A DI-{left
ins}S{down}"
62190 PRINT "PENSER (AVAILABLE AT EXTRA COST), WHICH{down}"
62200 PRINT "THE SUPERBAT MUST EAT TO STAY 'ALIVE'."
62210 GOSUB 60200
62220 PRINT "THE 'INSECTS' HAVE VARIOUS CALORIC{down}"
62230 PRINT "VALUES, AS FOLLOWS:{down}"
62240 PRINT "{rvs V off} - ELEPHANT BUG. 25 CALORIES{down}"
62250 PRINT "{X} - BUFFET BEETLE. 10 CALORIES{down}"
62260 PRINT "{S} - FEAST FLY. DOUBLES CALORIES{down}"
62270 PRINT "{*} - LINT. HALVES CALORIES{down}"
62280 PRINT "{A} - DIRT BUG. WORTHLESS{down}"
62300 PRINT "TO LIVEN THINGS UP, THE SUPERBAT IS{down}"
62310 PRINT "COATED WITH HIGH-RESILIENCY RUBBER, SO{down}"
62320 PRINT "THAT IT BOUNCES EXTREMELY WELL."
62340 GOSUB 60200
62350 PRINT "TO CONTROL THE SUPERBAT, YOU USE THE{down}"
62360 PRINT "NUMERIC PAD IN THE USUAL MANNER, SO THT{left
ins}A{down}"
62370 PRINT
      "{rvs}8{off} MEANS UP, {rvs}6{off} MEANS RIGHT, AND SO ON.{down}"
62380 PRINT "NOTE THAT PRESSING THE BUTTON MERELY{down}"
62390 PRINT "APPLIES FORCE. IT DOES NOT NECESSARILY{down}"
62400 PRINT "CAUSE AN IMMEDIATE CHANGE IN DIRECTION.{down}"
62410 PRINT "FOR EXAMPLE, IF THE SUPERBAT IS MOVING{down}"
62420 PRINT "LEFT, AND YOU HOLD DOWN {rvs}6{off} TO MOVE RIGHT,{left
ins}T{down}"
62430 PRINT "IT MUST FIRST SLOW DOWN AND STOP BEFORE{down}"
62440 PRINT "IT STARTS MOVING RIGHT."
62450 GOSUB 60200
62460 PRINT "TO APPLY A FORCE FOR A LONG TIME, JUST{down}"
62470 PRINT "HOLD THE BUTTON DOWN FOR A WHILE. YOU{down}"
62480 PRINT "DON'T NEED TO LET IT UP EXCEPT TO CHANE{left
ins}G{down}"
62490 PRINT "THE DIRECTION OF THE APPLIED FORCE.{down}"
62500 PRINT
      "ALSO, THE {rvs}1{off}, {rvs}3{off}, {rvs}7{off}, AND {rvs}9
      {off} KEYS DON'T DO{down}"
62510 PRINT "ANYTHING EXCEPT GET IN THE WAY.{down}"
62520 PRINT "{down}REMEMBER THAT GRAVITY WILL EVENTUALLY{down}"

```

```
62530 PRINT "PULL.THE.SUPERBAT.TO.THE.FLOOR.IF.YOU{down}"
62540 PRINT "DON'T.PUSH.IT.UP.OCCASIONALLY."
62550 GOSUB 60200
```



Your car is a white Porsche (shown as a solid white circle), and his is a black Ferrari (shown as a black circle with white border). The track has five lanes which are separated from each other by large steel slabs. In the four shaded areas, the slabs have been removed, allowing you to change lanes. Since your car is always moving, you need only change lanes by using the numeric pad in the usual fashion.

26

```

off ]}"
260 PRINT S$"{l rvs &&&&&&& off ]}+_+N+A+B+_+{l rvs &&&&&&&
off ]}"
270 PRINT S$"{l rvs &&&&&&& off ] rvs}GAME{&&}SCORE{off ] rvs
&&&&&&& off ]}"
280 PRINT S$"{l rvs &&&&&&& off ]}.....{l rvs &&&&&&&
off ]}"
290 PRINT S$"{l rvs &&&&&&& off }_+{-@@@@@@@@@@@@= right rvs &&&&&&&
off ]}"
300 PRINT
S$"{l}+{l}+{l}+{l}+{l}+_+_+{&&&&}+_+_+{l}+{l}+{l}+{l}+{l}"
310 PRINT S$"{l}_+{l}_+{l}_+{l}_+{-@@@@&&&&@@@@=}_+{l}_+{l}_+{l}_+{l}"
320 PRINT S$"{l}+{l}+{l}+{l}+_+_+_+{&&&&}+_+_+_+{l}+{l}+{l}+{l}+{l}"
330 PRINT S$"{l}_+{l}_+{l}_+{-@@@@@@@@&&&&@@@@@@@@=}_+{l}_+{l}_+{l}"
340 PRINT S$"{l}+{l}+{l}+_+_+_+_+{&&&&}+_+_+_+_+{l}+{l}+{l}+{l}"
350 PRINT S$"{l}_+{l}_+{-@@@@@@@@&&&&@@@@@@@@=}_+{l}_+{l}"
360 PRINT S$"{l}+{l}+_+_+_+_+_+{&&&&}+_+_+_+_+_+{l}+{l}"
370 PRINT S$"{l}_+{-@@@@@@@@@@@@&&&&@@@@@@@@@@@@=}_+{l}"
380 PRINT S$"{l}+_+_+_+_+_+_+{&&&&}+_+_+_+_+_+_+{l}"
390 PRINT S$"{-@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@= home}"
400 PRINT "{12°down}";TAB(15);MID$(STR$(TR),2)
410 CRT=32768
420 K=CRT+23*WD+15+LEN(S$):P=K+2
425 DF=1
430 D(0)=1:D(1)=-WD:D(2)=-1:D(3)=WD
440 DK=2:DP=0:TK=1:TP=1:V9=43
450 SK=0:SC=0:FP=V3:FK=102
460 POKE P1,0:POKE K,87:POKE P,81:IF P=K THEN 690
470 POKE P1,SD:FOR P9=1 TO 10:NEXT P9:POKE P1,0:SD=0
480 C=PEEK(P+D(DP)):IF C<>43 AND C<>V9 THEN 500
485 IF C=43 THEN SC=SC+10:SD=80:GOTO 495
490 SC=SC+20:SD=40
495 PRINT "{up}"TAB(18)RIGHT$("....."+STR$(SC),8):C=V8
497 IF SC>999 THEN V9=90
500 IF C=93 OR C=64 THEN DP=DP+1:C=V2:IF DP>3 THEN DP=0
510 IF C=87 THEN 690
520 POKE P,FP:FP=C:P=P+D(DP):IF RND(1)>.85 THEN 600
530 C=PEEK(K+D(DK)):IF C=93 OR C=64 THEN DK=DK-1:C=V2:IF DK<0 THEN
DK=3
540 IF C=V8 THEN C=V9
545 IF C=81 THEN 690
550 POKE K,FK:K=K+D(DK):FK=C
552 IF C<>V3 AND C<>V4 THEN TL=DF:GOTO 600
556 TL=TL-1:IF TL<0 THEN 600
560 IF C=V3 AND TK<TP THEN K=K+V7*D(DK):TK=TK+V1
570 IF C=V3 AND TK>TP THEN K=K-V7*D(DK):TK=TK-V1
580 IF C=V4 AND TK<TP THEN K=K-2*D(DK)/WD:TK=TK+V1
590 IF C=V4 AND TK>TP THEN K=K+2*D(DK)/WD:TK=TK-V1
600 IF FP<>V3 AND FP<>V4 THEN 680
601 POKE QP,255:GET K$:IF K$="" THEN K$="5"
603 IF K$="Q" OR K$="_" THEN 780

```

```

605 KB=ASC(K$)-48:MV=0
610 IF KB<2 OR KB>8 OR (KB AND 1) THEN 680
620 IF KB=8 AND FP=V3 THEN JS=P-WD*2:TT=TP+SGN(D(DP)):MV=1
630 IF KB=2 AND FP=V3 THEN JS=P+WD*2:TT=TP-SGN(D(DP)):MV=1
640 IF KB=4 AND FP=V4 THEN JS=P-2:TT=TP-SGN(D(DP)):MV=1
650 IF KB=6 AND FP=V4 THEN JS=P+2:TT=TP+SGN(D(DP)):MV=1
655 IF MV=0 THEN 680
660 IF MV THEN SD=151
663 IF ABS(P-JS)<>2 AND ABS(P-JS)<>WD*2 THEN STOP
670 IF PEEK(JS)=FP THEN TP=TT:P=JS
680 DF=1+INT(SC/500)
685 GOTO 460
690 X=P-CRT:PRINT "{home}";
700 Y=INT(X/WD):X=X-Y*WD
710 FOR A=1 TO Y:PRINT "{down}";:NEXT
720 PRINT TAB(X-2)"{right CV}-{up 3°left M}N 2°down 3°left N}M
    home 3°down}"
730 POKE P2,15
740 FOR A=1 TO 25:POKE P1,255:POKE P1,250:POKE P1,253:NEXT
750 POKE P1,0:POKE P3,16:POKE P2,15
760 TM=TI+150
770 IF TI<TM THEN 770
780 GET A$:IF A$<>" " THEN 780
790 IF SC>HS THEN HS=SC
800 CS=CS+SC
810 PRINT "{clr 5°down}_YOUR_SCORE_FOR_GAME"TR"WAS"SC;"{left},"
820 PRINT
    "{down}_FOR_AN_AVERAGE_OF"INT(CS/TR+.5)"POINTS/GAME.{down}"
830 PRINT "_YOUR_HIGH_SCORE_SO_FAR_IS"HS;"{left}."
840 PRINT "{3°down}_DO_YOU_WISH_TO_PLAY_AGAIN?_";:GOSUB 60000
850 A$=LEFT$(IN$,1)
860 IF A$="N" THEN PRINT "{clr}":GOTO 880
870 FOR A=1 TO 200:NEXT:TR=TR+1:GOTO 140
880 POKE P2,0:POKE P3,0:POKE 59468,12
890 END
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
    ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
    GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
    Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
    GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
    NEXT Z:GOTO 60000

```

```

60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{TO}";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;"_{-}_CURSOR_{#}";NM$;
    "_{(C)}_{1981}_{PAGE}";PN
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}";GOSUB 60400:GOTO 100
60400 CRT=32768:CR$=CHR$(13):WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60420 QP=515:QV=1:CR$=CHR$(13)
60430 IF PEEK(50000) THEN QP=151:QV=2
60440 RETURN
60500 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;
    "_{BY}_{MALCOLM}_{MICHAEL}"
62010 PRINT "{down}_{CURSOR}_{#16}_{COPYRIGHT}_{(C)}_{1981}"
62020 GOSUB 60500
62030 PRINT "{down}_{A}_{RACE}_{AGAINST}_{DESTRUCTION}..."
62080 PRINT "{2°down}PRESS_{rvs}RETURN{off}_{TO}_{BEGIN}"
62090 GET T$:IF T$="" THEN 62090
62100 GOTO 60300

```



PUZZLES



HANOI

By Glen Fisher

Somewhere, secluded from modern civilization, a group of priests has been moving golden disks from one great golden pedestal to another. When they're done, so the story goes, the world will end.

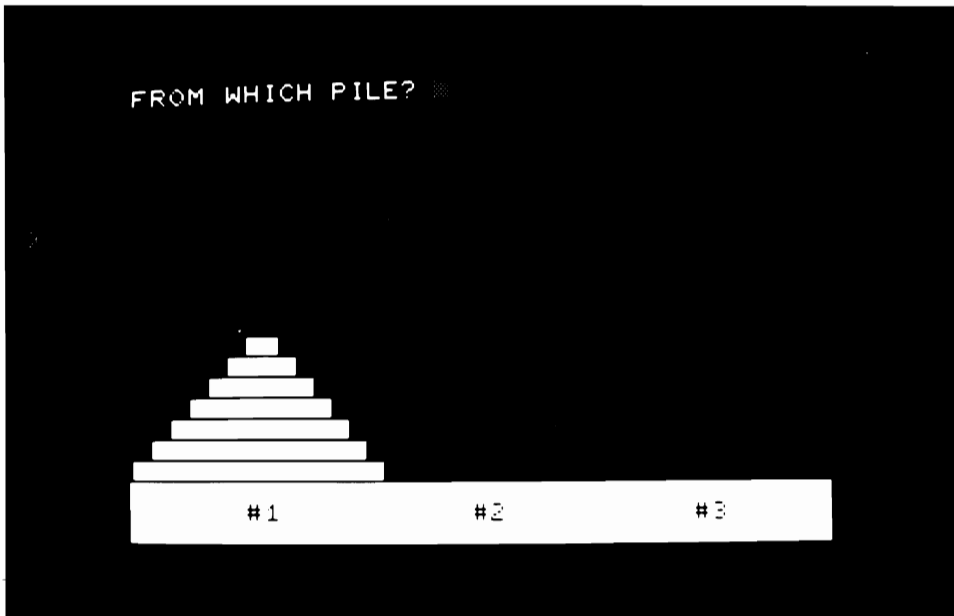
In the legend, there were three pedestals and, some say, 64 disks in a regular series (each disk slightly larger than the last). Initially, all the disks were on one of the three pedestals (with the largest at the bottom and the smallest at the top). The priests moved the whole pile to another pedestal, one disk at a time. There was one restriction placed on them: a disk could be placed only on top of a larger disk, never on a smaller one. With that restriction, assuming they could move one disk per *second*, it would take them more

than a half trillion *years* to move the pile of 64 disks.

HANOI is a much-simplified version of that task, with a much less cosmic ending; when you finish, only the program ends. However, the restrictions are the same. You may choose from two to seven disks in the pile. To make a move, just indicate from which pile you want to take the top disk, and on which pile you want the disk placed. HANOI won't allow you to make an illegal move.

If you want to give up, type "Q" (for quit). When you're done, you'll be told how many moves you took, as well as the minimum number of moves necessary.

Oh yes, the program is called HANOI because the priests were once thought to be in Hanoi.

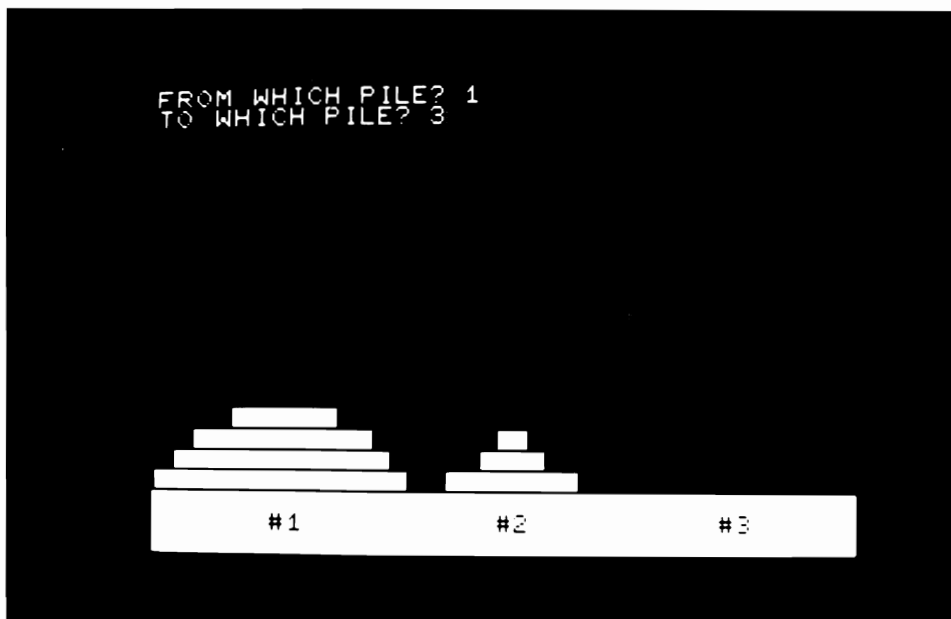


```
0 CLR:POKE 59468,12:NM$="5"
6 REM HANOI COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81 RJ
90 GOSUB 62000
100 T$="{off}_.{off $ off / off 9 off " rvs 8 rvs 7 rvs #}"
105 B$="{rvs $ rvs / rvs 9 rvs " off 8 off 7 off # off}_"
110 DIM TP$(8),T$(8),BT$(8),B$(8)
115 CR$=""
120 FOR I=1 TO 8
130 C$=MID$(T$,2*I,1):C$=C$+C$+C$:C$=C$+C$+C$:C$=LEFT$(C$+C$,14)
```

```

140 TP$(I)=MID$(T$,2*I-1,1)+C$
150 C$=MID$(B$,2*I,1):C$=C$+C$+C$:C$=C$+C$+C$:C$=LEFT$(C$+C$,14)
160 BT$(I)=MID$(B$,2*I-1,1)+C$
170 NEXT I
200 L$="{rvs}_ {rvs 4 rvs ! off * off}_ "
210 R$="{off}_ {off 4 off ! rvs * rvs}_ "
220 DIM LF$(5),RT$(5)
230 FOR I=1 TO 5:LF$(I)=MID$(L$,2*I-1,2):RT$(I)=MID$(R$,2*I-1,2):
NEXT I
500 PRINT :PRINT "HOW MANY DISKS (UP TO 7) ? ";:GOSUB 60000:PRINT
510 IF IN$="" THEN PRINT "LET'S USE 3":IN$="3":FOR I=1 TO 500:NEXT
520 FOR I=1 TO LEN(IN$)
530 :T$=MID$(IN$,I,1):IF T$>="0" AND T$<="9" THEN 550
540 PRINT "NO ";T$;"'S PLEASE.":GOTO 500
550 NEXT I
560 N=VAL(IN$):IF N>7 THEN PRINT "I CAN'T HANDLE MORE THAN 7.":
GOTO 500
570 IF N<2 THEN PRINT "DON'T BE RIDICULOUS!":GOTO 500
900 DN$="{home 20°down}"
910 RT$="{14°right}"
1000 FOR I=1 TO 3:FOR J=0 TO 7:P(I,J)=0:NEXT J:NEXT I
1100 PRINT "{clr}";DN$;:P(1,0)=N

```



```

1110 FOR I=1 TO 3:
PRINT "{rvs}.....{left
ins}_":NEXT I
1120 PRINT DN$;"{down rvs 6°right}#1{11°right}#2{11°right}#3"
1150 PRINT DN$;LEFT$("{7°up}",N);
1160 FOR I=1 TO N
1170 PRINT LEFT$("{off}"+RT$,7-I+1);LEFT$("{rvs L}"+BT$(1),2*I+1);
"{:}"
1180 P(1,N-I+1)=I*2
1190 NEXT I

```

```

1800 MV=0
1900 TM=TI+60
1910 IF TI<TM THEN 1910
1950 GOSUB 8000
2000 PRINT :PR$="FROM WHICH PILE?":GOSUB 5000:IF IN$="" THEN 1950
2010 F=VAL(IN$)
2020 IF P(F,0)<1 THEN PRINT "THAT PILE IS EMPTY.":GOTO 1900
2050 PR$="TO WHICH PILE?":GOSUB 5000:IF IN$="" THEN 1950
2060 T=VAL(IN$)
2070 IF F=T THEN PRINT "THAT MOVE DOES NOTHING.":GOTO 1900
2080 IF P(T,0)=0 THEN 2100
2090 IF P(F,P(F,0))>P(T,P(T,0)) THEN PRINT "IT WON'T FIT.":GOTO 1900
2100 FC=P(F,0):FW=P(F,FC):TC=P(T,0):TW=P(T,TC):MV=MV+1
2110 X=1+13*(F-1)+7-FW/2
2120 PRINT DN$;LEFT$("{9°up}",FC+1);LEFT$("{off}"+RT$+RT$+RT$,X);
2125 LF$=LEFT$("{down 14°left}",FW+1)
2130 FOR I=1 TO 8:T$(I)=LEFT$(TP$(I),1+FW):B$(I)=LEFT$(BT$(I),1+FW):
    NEXT I
2135 HT=0:J=1:IF F>T THEN J=-1
2140 FOR I=F TO T STEP J
2145 IF P(I,0)>HT THEN HT=P(I,0)
2150 NEXT I
2155 IF HT=P(T,0) THEN HT=HT+1:GOTO 2165
2160 IF ABS(F-T)>1 THEN IF HT=P(2,0) THEN HT=HT+1
2165 FOR I=FC TO HT
2170 :FOR J=1 TO 8
2175 :PRINT T$(J);LF$;B$(J);LF$;"{2°up}";
2180 :NEXT J
2185 :PRINT "{up}";
2190 NEXT I
2195 PRINT "{down}";
2200 L$=LEFT$("{16°left}",1+FW):R$=LEFT$(RT$,FW-1)
2201 PRINT "{rvs}";LEFT$(".....",FW);"{off}";
    LEFT$("{16°left}",FW);
2205 IF F>T THEN 2250
2210 FOR I=F*13 TO T*13-1
2215 :FOR J=1 TO 5
2220 PRINT LF$(J);R$;RT$(J);L$;
2225 NEXT J
2226 PRINT "{right}";
2230 NEXT I
2235 GOTO 2300
2250 FOR I=T*13 TO F*13-1
2255 PRINT "{left}";
2260 :FOR J=5 TO 1 STEP -1
2265 ::PRINT LF$(J);R$;RT$(J);L$;
2270 :NEXT J
2275 NEXT I
2300 FOR I=HT-1 TO TC STEP -1
2310 :FOR J=8 TO 1 STEP -1
2320 :PRINT T$(J);LF$;B$(J);LF$;"{2°up}";

```



```

60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN GOTO 60250
60205 GOSUB 60500
60210 PRINT "{down}PRESS_{rvs}RETURN{off}_{TO}";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;".-CURSOR.#";NM$;
      ".(C).1981....PAGE";PN;"{down}"
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}":CLR:CR$=CHR$(13):GOTO 100
60430 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PG$="HANOI":LP=1:CR$=CHR$(13)
62020 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;".BY.GLEN.FISHER"
62021 PRINT "{down}....CURSOR.#5..COPYRIGHT.(C).1981"
62022 GOSUB 60500
62025 PRINT "{down}ANIMATED.TOWERS.OF.HANOI.PUZZLE"
62030 PRINT "{2°down}WANT.DIRECTIONS?";:GOSUB 60000
62040 IF LEFT$(IN$,1)="Y" THEN GOTO 62060
62050 PRINT "{clr 3°down}":RETURN
62060 GOSUB 60200
62070 PRINT "THE.OBJECT.OF.THIS.PUZZLE.IS.TO.MOVE"
62080 PRINT "A.PILE.OF.DISKS..HOWEVER,.A.LARGER.DISK"
62090 PRINT "MAY.NEVER.BE.PLACED.ON.TOP.OF.A.SMALLER"
62100 LP=1:PRINT "DISK."
62110 PRINT "{down}THE.PET.WILL.ASK.YOU.HOW.MANY.DISKS"
62120 PRINT "YOU.WISH.TO.WORK.WITH..(BEGINNERS:"
62130 PRINT "TRY{.}3.OR.4...)..IT.WILL.THEN.DRAW"
62140 PRINT "A.PICTURE.OF.THE.DISKS,.AND.ASK:"
62150 PRINT "'FROM.WHICH.PILE?'..ENTER.THE.NUMBER"
62160 PRINT "OF.THE.PILE.FROM.WHICH.YOU.WANT.TO"
62170 PRINT "TAKE.THE.TOP.DISK..THEN,.YOU.ENTER"
62180 PRINT "THE.NUMBER.OF.THE.PILE.WHERE.YOU"
62190 PRINT "WANT.TO.PLACE.THAT.DISK."
62200 PRINT "{down}REMEMBER.-.YOU.ALWAYS.MOVE.ONE.DISK"
62210 PRINT "AT.A.TIME!.(THE.PET.WON'T.LET.YOU.MAKE"
62220 PRINT "ILLEGAL.MOVES)."
62230 GOSUB 60200

```

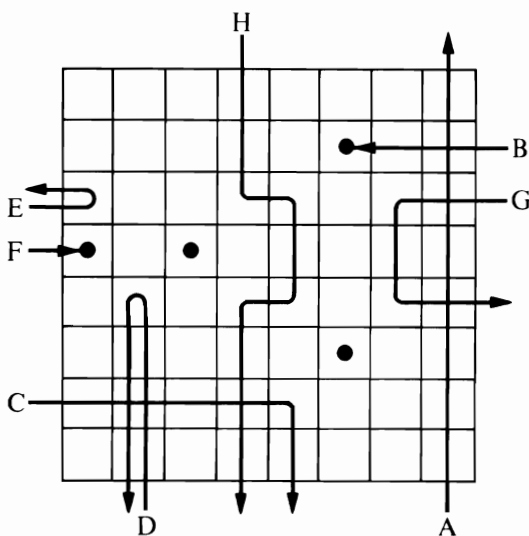


BOX

By Howard Arrington

The semiconductor industry is always looking for new materials with which to make its ever-smaller products. As you may know, integrated circuit chips are made by taking very pure silicon and adding very specific impurities in very particular patterns. In BOX, you have a small piece of a silicon chip, and must determine the exact position, within an 8×8 area, of the atoms from the added impurity. Naturally, since atoms are far too small to see, you must use some other tool to find them.

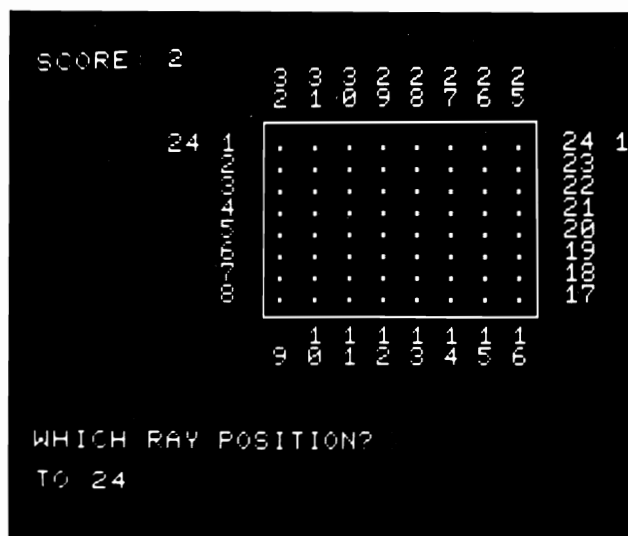
Your tools in this game are a particle "gun" (which shoots a high-energy ray into the chip) and a set of detectors to tell where the ray comes out again, if it does. Since the silicon is so pure to start with, it forms very perfect crystals. As a result, the number of ways a ray can interact with an impurity atom are few, and are shown in the diagram below. The drawing shows a sample 8×8 board, with the atoms marked, and several rays shot into it.



- A) If a shot hits nothing, it goes straight through and comes out the other side.
- B) If a shot hits an atom straight on, it is absorbed by the nucleus, and vanishes. This is a *hit*.
- C) A shot that passes next to an atom is *deflected* 90 degrees away from the atom. Note that it never gets to be even with the atom.

- D) A shot aimed between two atoms one space apart is *reflected* back to its starting point. (It gets deflected 90 degrees twice, resulting in a 180-degree turnaround.)
- E) A shot just to one side of an atom (at the edge) is also *reflected*.
- F) A shot directly at an atom (at the edge) is a *hit*, just as any other head-on shot.
- G) Shots may interact with more than one atom. Here is a shot that gets deflected twice, and comes out the same side it came in.
- H) This is another multiple interaction. However, the results of this one appear the same as no interaction at all. The shot *appears* to go straight through, even though it is deflected by three atoms (twice by one of them).

When the game starts, you'll be asked how many atoms (from 1 to 16) should be put into the



8×8 area. You must then find them by shooting rays.

To make a shot, type in the number of the spot where you want the ray to enter. The result will be shown next to the entry spot by the following: an "H" for a hit, an "R" for a reflection, and the number of the exit spot for anything else. (In the last case, the exit spot will be marked with the

number of the entry spot, as well.)

When you've figured out where an atom is, you should mark it. To do so, press [RETURN] when asked for an entry spot. The PET will ask you for the row and column of the atom (as with shots, use the numbers along the edge). If you're right, the atom will be marked. If you're wrong, the PET will say so. The game is over when

you've marked all the atoms.

To quit and to find out where the atoms are, enter [0] (zero) for the entry point of a shot. A score is being kept during all of this. You are given one point for each hit or reflection, two points for any other result, and five points for each wrong guess of an atom's location. The lower the score, the better you are.

```

0 CLR:POKE 59468,12:PG$="BOX":GOSUB 62000
6 REM COPYRIGHT(C)1981 THE CODE WORKS
100 GOSUB 60400:REM 10/6/81
102 DIM L(32):FOR I=0 TO 7:L(I+1)=I*WD:NEXT I:FOR I=0 TO 7:
    L(I+9)=I*2+WD*12+5:NEXT I
105 FOR I=0 TO 7:L(I+17)=WD*7+25-I*WD:NEXT I:FOR I=0 TO 7:
    L(I+25)=- (WD*5)+19-I*2:NEXT I
110 DATA 487,489,491,493,495,497,499
115 DATA 305,265,225,185,145,105,65,25,-181,-183,-185,-187,-189,
    -191,-193,-195
125 DEF FN R(Z)=INT(8*RND(1)+1)
135 CR$=CHR$(13):A=CRT+WD*6+7
137 W$="....."
140 PRINT :PR$="{up}HOW MANY ATOMS?":GOSUB 4000:IF IN$<>" THEN
    N=IN:GOTO 150
145 PRINT "I'LL ASSUME FIVE":N=5
150 K=0:HA=N:S=0
151 H$="{home 20°down}"
155 RS=22*WD+CRT
160 IF N<1 THEN 3500
161 IF N>16 THEN PRINT "16 IS MAXIMUM{up 2°right}"W$"{up}":GOTO 140
165 FOR J=0 TO 9:FOR I=0 TO 9:B(I,J)=0:NEXT I:NEXT J
175 FOR I=1 TO N
180 X=FN R(1):Y=FN R(1)
185 IF B(X,Y)<>0 THEN 180
190 B(X,Y)=1:NEXT I
195 GOSUB 1000
200 GOSUB 2000:IF K<>Z THEN R=K:K=Z:Z=R:GOSUB 2000
201 PRINT H$"{15°right}"W$
205 PR$="{up}WHICH RAY POSITION?":GOSUB 4000:IF IN$="" THEN IN=-1
206 PRINT W$:R=IN:K=R:IF R<1 THEN 3000
207 PRINT W$;"{up}"
208 IF R>32 THEN 201
210 ON (R-1)/8+1 GOTO 250,260,270,280
220 PRINT "ERROR.....":GOTO 200
250 X=0:Y=R:U=1:V=0:GOTO 300
260 X=R-8:Y=9:U=0:V=-1:GOTO 300
270 X=9:Y=25-R:U=-1:V=0:GOTO 300
280 X=33-R:Y=0:U=0:V=1
300 X1=X+U:Y1=Y+V
310 IF U=0 THEN X2=X1-1:X3=X1+1:Y2=Y1:Y3=Y1:GOTO 330
320 Y2=Y1-1:Y3=Y1+1:X2=X1:X3=X1
330 ON 8*SGN(B(X1,Y1))+SGN(B(X2,Y2))+2*SGN(B(X3,Y3))+1 GOTO 400,
    410,420,410

```



```

340 S=S+1:PRINT "HIT.....":Z=K:GOTO 200
400 X=X1:Y=Y1:GOTO 500
410 Z=1:GOTO 450
420 Z=-1
450 IF U=0 THEN U=Z:V=0:GOTO 500
460 U=0:V=Z
500 ON (X+15)/8 GOTO 550,520,560
510 STOP
520 ON (Y+15)/8 GOTO 570,300,580
530 STOP
550 Z=Y:GOTO 600
560 Z=25-Y:GOTO 600
570 Z=33-X:GOTO 600
580 Z=8+X
600 IF Z=R THEN PRINT "REFLECTION":S=S+1:GOTO 200
610 S=S+2:PRINT "TO";Z;"{left}.....":GOTO 200
700 A=A-WD+4:D1=87
710 FOR J=1 TO 8:FOR I=1 TO 8
720 IF B(I,J)<>1 THEN 740
730 T=A+I*2+J*WD:POKE T,D1
740 NEXT I:NEXT J
750 PRINT H$;W$
760 PRINT W$:PRINT W$;"{3°up}":GOTO 135
1000 PRINT "{clr 3°down}.....3.3.3.2.2.2.2.2"
1020 PRINT TAB(13);"2.1.0.9.8.7.6.5"
1030 PRINT TAB(12);"{0000000000000000.}"
1040 FOR I=1 TO 8
1050 PRINT ".....";RIGHT$(STR$(I),2);"_{1}}.....{1}_";
1055 PRINT RIGHT$(STR$(25-I),2)
1060 NEXT I
1070 PRINT TAB(12);"{-0000000000000000=}"
1080 PRINT TAB(13);"..1.1.1.1.1.1.1"
1090 PRINT TAB(13);"9.0.1.2.3.4.5.6"
1100 RETURN
2000 PRINT "{home 2°down}SCORE: ";S;"{left}.."
2005 IF K=0 THEN RETURN
2010 X=PEEK(RS):ZZ=L(K)+A
2020 IF X=20 THEN X$=RIGHT$(STR$(Z),2):GOTO 2100
2030 POKE ZZ+1,X:RETURN
2100 D1=ASC(MID$(X$,1,1)):D2=ASC(MID$(X$,2,1))
2105 IF K>8 THEN 2140
2110 POKE ZZ,D1:POKE ZZ+1,D2:RETURN
2140 IF K>16 THEN 2200
2148 IF D1=32 THEN D1=D2:D2=32
2150 POKE ZZ+1,D1:POKE ZZ+WD+1,D2:RETURN
2200 IF K>24 THEN 2300
2208 IF D1=32 THEN D1=D2:D2=32
2210 POKE ZZ+1,D1:POKE ZZ+2,D2:RETURN
2300 POKE ZZ-WD+1,D1:POKE ZZ+1,D2:RETURN
3000 IF R=0 THEN 700
3005 PRINT H$;W$

```

```

3006 PRINT W$:PRINT W$:PRINT W$;"{3°up}"
3010 PR$="{up}WHICH_ROW?_":GOSUB 4000:IF IN$="" THEN PRINT :
    GOTO 3060
3012 I=IN:IF I>16 AND I<25 THEN I=25-I
3015 IF I<1 OR I>8 THEN PRINT "{up 7°right}"W$:GOTO 3010
3016 PRINT
3020 PR$="{up}WHICH_COLUMN?_":GOSUB 4000:IF IN$="" THEN 3060
3021 J=IN:IF J>8 AND J<17 THEN J=41-J
3025 IF J<25 OR J>32 THEN PRINT "{up 9°right}"W$:GOTO 3020.
3030 J=33-J
3035 IF J<1 OR J>8 THEN 3010
3040 IF B(J,I)=1 THEN POKE A-WD+4+J*2+I*WD,81:B(J,I)=2:HA=HA-1:
    GOTO 3060
3045 IF B(J,I)<>0 THEN PRINT "THAT_ONE_AGAIN???{up}":GOTO 3055
3050 PRINT "WRONG!{up}":S=S+5
3055 FOR I=0 TO 1500:NEXT I
3060 PRINT "{2°up}";W$:PRINT W$:PRINT W$
3065 IF HA=0 THEN 700
3070 K=0:Z=0:GOTO 200
3500 PRINT "{clr}":END
4000 PRINT PR$;:GOSUB 60000:IF IN$="" THEN RETURN
4010 FOR Z=1 TO LEN(IN$):T$=MID$(IN$,Z,1)
4020 IF T$=>"0" AND T$=<"9" THEN NEXT Z:IN=VAL(IN$):RETURN
4030 PRINT "NO_";T$;"'S_PLEASE.":PRINT "{2°up}"W$ LEFT$(W$,15):
    GOTO 4000
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20):PRINT "._._{3°left}";
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_&","ZC,1);"{left}";:ZC=3-ZC:
    ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
    GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
    Z$=CHR$((Z+128) AND 255)
60090 IF ZL>18 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
    GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
    NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "{down}PRESS_{rvs}RETURN{off}_TO_";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN CLR:PRINT "{clr}";:GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_CURSOR_#6";NM$;
    "._.(C).1981....PAGE";PN;"{down}"

```

```

60265 GOSUB 60500
60270 RETURN
60400 CRT=32768:PRINT "{clr}"
60410 WD=40:POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:
PRINT CHR$(142)
60415 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;
"._BY._HOWARD._ARRINGTON"
62010 PRINT "{down}.....CURSOR.#6...COPYRIGHT.(C).1981"
62013 GOSUB 60500
62015 PRINT "{down}FIND._THE._HIDDEN._ATOMS{down}"
62020 PRINT "{down}WANT._INSTRUCTIONS?._";:GOSUB 60000:IF LEFT$(IN$,
1)="Y" THEN 62040
62030 PRINT "{clr}":RETURN
62040 LP=4:GOSUB 60200
62100 PRINT "YOU._WILL._SHORTLY._SEE._AN._8X8._BOX._IN"
62110 PRINT "WHICH._SEVERAL._ATOMS._ARE._HIDDEN._.YOUR"
62120 PRINT "TASK._IS._TO._FIND._THE._ATOMS._.YOU._HAVE._AT"
62130 PRINT "YOUR._DISPOSAL._A._RAY._PROJECTER,_WHICH"
62140 PRINT "YOU._CAN._USE._TO._SEND._RAYS._INTO._THE._BOX"
62150 PRINT "FROM._ANY._OF._32._SPOTS._ALONG._THE._SIDES"
62160 PRINT "OF._THE._BOX._"
62165 PRINT "{down}ANY._GIVEN._RAY._CAN._MEET._ONE._OF._THREE"
62170 PRINT "FATES:._IT._MAY._HIT._AN._ATOM,_AND._BE"
62180 PRINT "ABSORBED;._IT._MAY._BE._REFLECTED,_AND._COME"
62190 PRINT "OUT._WHERE._IT._WENT._IN;._OR._IT._MAY._BE"
62200 PRINT "DIVERTED._AND._COME._OUT._SOMEWHERE._ELSE._"
62210 PRINT "YOU._WILL._BE._TOLD._WHAT._HAPPENS,_AND"
62220 PRINT "WHERE._THE._RAY._COMES._OUT._(IF._IT._DOES)._ "
62230 PRINT "{down}ON._THE._NEXT._PAGE,_YOU'LL._SEE._THE._RULES"
62235 PRINT "CONTROLLING._THE._PATH._OF._A._RAY._"
62240 GOSUB 60200
62250 PRINT "{down}1:._A_{._}RAY._THAT._HITS._AN._ATOM._DEAD._ON._IS"
62260 PRINT "._.ABSORBED,_AND._NEVER._COMES._OUT._.{}_{.}"
62265 PRINT TAB(34);"{@=}_{.-@}"
62270 PRINT "2:._A._RAY._THAT._HITS._AN._ATOM._TO._ONE._.O"
62280 PRINT "._.SIDE._IS._DIVERTED._90._DEGRESS._TO._{@._}_{0@}"
62290 PRINT "._.THAT._SIDE._";TAB(35);"{}_{.}"
62300 PRINT "{down}3:._A._RAY._THAT._PASSES._BETWEEN._TWO._.O._.O"
62310 PRINT "._.ATOMS._ONE._SQUARE._APART._IS._RE-._.{}_{.}"
62315 PRINT "._.FLECTED,_AND._RETRACES._ITS._PATH._.{}_{.}"
62320 PRINT "{down}4:._A._RAY._THAT._ENTERS._ALONGSIDE._.O{}_{.}"
62330 PRINT "._.AN._ATOM._IS._REFLECTED._.{}_{.}"
62340 PRINT "{down}5:._AT._ALL._OTHER._TIMES._A._RAY._FOLLOWS._A"
62350 PRINT "._.STRAIGHT._PATH._"
62360 GOSUB 60200
62370 PRINT "THE._PLACES._YOU._CAN._SEND._IN._A._RAY._ARE"
62380 PRINT "NUMBERED._FROM{right}1._TO_{._}32._.TO._SEND._A._RAY"
62390 PRINT "IN,_TYPE._THE._NUMBER._OF._THE._PLACE._YOU"
62400 PRINT "WANT._IT._TO._ENTER._AT._.IT._WILL._START._OUT"

```

```
62410 PRINT "TRAVELLING_VERTICALLY_OR_SIDEWAYS,"
62420 PRINT "DEPENDING_ON_WHICH_SIDE_IT_STARTS_AT."
62430 PRINT "{down}YOU_WILL_BE_TOLD_WHETHER_THE_RAY_HITS"
62440 PRINT "AN_ATOM,{ }IS_REFLECTED,_OR_COMES_OUT."
62450 PRINT "SHOULD_IT_COME_OUT,_YOU'LL_BE_TOLD"
62460 PRINT "WHERE...TO_GIVE_UP_(AND_FIND_OUT_WHERE"
62470 PRINT "THOSE_LOUSY_ATOMS_WERE),_TYPE_A_0"
62480 PRINT "(ZERO)_AS_THE_RAY_ENTRANCE._TO_GUESS"
62490 PRINT "AN_ATOM'S_LOCATION,_JUST_HIT_{rvs}RETURN{off}."
62500 GOSUB 60200
62510 PRINT "YOU_ARE_GIVEN_A_SCORE_ACCORDING_TO_HOW"
62520 PRINT "WELL_YOU_DO._FOR_EACH_RAY_THAT_GOES_IN"
62530 PRINT "YOU_GET_ONE_POINT._IF_IT_COMES_OUT"
62540 PRINT "AGAIN,{ }YOU_GET_ANOTHER_POINT._FOR_EACH"
62550 PRINT "WRONG_GUESS_YOU_MAKE,_YOU_GET_FIVE"
62560 PRINT "POINTS...A_LOW_SCORE_INDICATES_HIGH"
62570 PRINT "SKILL."
62630 PRINT "{2°down}.....HAVE_FUN!!{2°down}"
62640 GOSUB 60200
```



MIND

By Glen Fisher

Mind is our version of the popular game “Master Mind”¹, which is a game played with colored pegs on a board. One player invents a color pattern, and the other player tries to guess the pattern. In the computer version, since we can’t use actual pegs, we just print the color names on the screen. The PET will choose the color pattern, and it is your job to discover the pattern.

You enter your guess using the [R], [O], [Y], [G], [B], and [V] (for Red, Orange, Yellow, Green, Blue, and Violet) keys. As you press them, the appropriate color names will be printed on the screen. At its simplest, entering your guess involves just pressing four keys, and then [RETURN]. If you press the wrong key, just press [DEL] (or [LEFT]) to back up the cursor, and press the key you intended. [SPACE] or [RIGHT] move the cursor to the right. Finally, the [?] key means “I give up. What’s the answer?”

Once you’ve pressed [RETURN], the PET will tell you how many “hits” or “misses” you got. You get a *hit* when a color in your guess matches (in color *and* position) a color in the pattern. For example:

pattern: R G B V

guess: B G R O

For that guess, you would get one hit, for the green. Neither red nor blue would be a hit, because they’re in the wrong places. You get a *miss* when one of the colors in your guess matches one of the colors in the pattern, but is in the wrong place. In the above example, red and blue would be misses, since the pattern has both red and blue in it, but in your guess they’re not where they are in the pattern.

There are two complications. First, the PET only tells you *how many* hits and misses you got. It doesn’t say which colors in your guess gave you the hits or misses. In the above example, you would merely be told “one hit, two misses,” with no clue as to which colors were involved. It is

nonetheless possible to figure out the pattern. You make several guesses, and figure out what pattern of colors could produce the hits and misses you’ve been getting. For example, with the pattern above, if you change your guess to

B O R G

you will get “0 hits, 3 misses.” You now know that either Green or Orange was the hit, because they’re the only colors you changed, and the hit went away.

The other complication is that the same color can be used several times in the pattern. Thus, the pattern could be R O G R, with two Reds in it. The thing to be aware of is that any given peg in the guess can make a hit or a miss, but never both. Likewise with the pattern, one peg can only be one hit or miss. For example, with the pattern R O G R, a guess of R Y B O would give you one hit (for red) and one miss (for orange). The other red wouldn’t count.

For a guess of R Y R B, you’d also get one hit (for red) and one miss (for the *other* red). A guess of Y R G B would give a hit (for green) and a miss (for red). You only get one miss because you only guessed one red. A guess of G R B G would give no hits and two misses, one for red, and one for green. There’s only one green in the pattern, so only one of your greens is a miss. The other green is ignored.

To show you how many hits and misses you got, the PET prints dots beside your guess. A white dot marks a hit and a black dot marks a miss. The location of the dots has nothing to do with where the hits or misses are. The PET just prints all the white dots, followed by all the black dots. If you get no dots at all, none of the colors in your guess are in the pattern.

Finally, if you’re feeling extra sharp, you may put a limit on the number of guesses allowed on a pattern. Exceed the limit and the PET tells you the answer, and marks that game as a loss for you. If you don’t want a limit, just press [RETURN] in answer to the question.

¹ “Master Mind” is a registered trademark of Invicta Plastics (U.S.A.) Ltd.

GAME	1	2	3	4
1	YELLOW	YELLOW	YELLOW	YELLOW
2	BLUE	BLUE	BLUE	BLUE
3	GREEN	GREEN	GREEN	GREEN
4	RED	RED	RED	RED
5	ORANGE	ORANGE	RED	RED
6	VIOLET	VIOLET	VIOLET	RED
7	RED	VIOLET	RED	VIOLET
8	RED	VIOLET	VIOLET	VIOLET
9	VIOLET	VIOLET	RED	VIOLET

THAT WAS THE PATTERN!

CARE TO TRY AGAIN? ■

```

0 CLR:POKE 59468,12
6 REM MIND COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81RJ
90 PG$="MIND":NM$="7":GOSUB 62000
100 CR$=CHR$(13):PRINT CHR$(142)
105 PRINT "DO YOU WANT A LIMIT ON THE NUMBER OF":PRINT "GUESSES? ";
:GOSUB 60000
108 IF IN$="" THEN PRINT "{down}THERE WILL BE NO LIMIT, THEN.":
FOR I=1 TO 2000:NEXT
110 IN$=LEFT$(IN$,1):IF IN$="N" OR IN$="" THEN LM=1E38:GOTO 200
115 IF IN$<>"Y" THEN PRINT "PLEASE ANSWER 'YES' OR 'NO'.":GOTO 105
120 PRINT "WHAT SHOULD THE LIMIT BE? ";:GOSUB 60000
130 IF IN$="" THEN PRINT "I'LL ASSUME 10.":LM=10:GOTO 200
140 FOR I=1 TO LEN(IN$):T$=MID$(IN$,I,1)
150 IF T$<"1" OR T$>"9" THEN PRINT "NO. ";T$;"S, PLEASE.":GOTO 120
160 NEXT I:LM=VAL(IN$)
200 READ T$:IF T$<>" " THEN C=C+1:GOTO 200
205 DATA RED,ORANGE,YELLOW,GREEN,BLUE,VIOLET,
210 N=4:CZ=6:B$="":FOR I=1 TO CZ:B$=B$+" ":L$=L$+"{@" :NEXT I
220 DIM P(N),G(N),PS(N),C$(C)
230 RESTORE:FOR I=1 TO C:READ C$:C$(I)=LEFT$(C$+B$,CZ):NEXT I:
C$(0)=B$
240 H$="{off}":M$="{off}":FOR I=1 TO N:H$=H$+"{Q}":M$=M$+"{W}":
NEXT I
500 TB=6:DEF FN T(N)=(CZ+1)*(N-1)+TB
1000 FOR I=1 TO N:P(I)=INT(RND(1)*C)+1:NEXT I
1010 PRINT "{clr}";:FOR I=1 TO N:PRINT TAB(FN T(I)+(CZ/2)-1);I;:
NEXT I
1020 PRINT :FOR I=1 TO N:PRINT TAB(FN T(I));L$;:NEXT I
1030 GM=GM+1:TN=0:PRINT "{home}GAME":PRINT GM
1100 IF TN>=LM THEN 1960
1110 TN=TN+1:TT=TT+1:PRINT TAB(TB-3);RIGHT$(STR$(TN),2):T=1:OT=T
1120 FOR I=1 TO N:G(I)=0:NEXT I

```

```

1200 IF OT<>T THEN PRINT "{up}";TAB(FN T(OT));C$(G(OT))
1210 IF T<1 THEN T=N
1220 IF T>N THEN T=1
1230 PRINT "{up}";TAB(FN T(T));"{rvs}";C$(G(T)):OT=T
1300 GET T$:IF T$="" THEN 1300
1310 IF T$="." OR T$="{right}" THEN T=T+1:GOTO 1200
1320 IF T$=CHR$(20) OR T$="{left}" THEN T=T-1:GOTO 1200
1330 IF T$=CHR$(13) THEN 2000
1340 FOR I=1 TO C
1350 :IF T$=MID$("123456789",I,1) OR T$=LEFT$(C$(I),1) THEN 1900
1360 NEXT I
1370 IF T$=">" THEN T2=T+1:GOTO 1800
1380 IF T$="<" THEN T2=T-1:GOTO 1800
1390 IF T$="?" THEN 1950
1400 GOTO 1300
1800 IF T2>N THEN T2=1
1810 IF T2<1 THEN T2=N
1820 I=G(T):G(T)=G(T2):G(T2)=I
1830 OT=T2:GOTO 1200
1900 G(T)=I:T=T+1
1910 GOTO 1200
1950 PRINT "{up}....."
1955 PRINT TAB(TB+6);"THE ANSWER IS:{down}":H=0:GOTO 1990
1960 PRINT TAB(TB);"{down}..SORRY,THE SOLUTION WAS:{down}":H=0:
GOTO 1990
1990 LS=LS+1:TT=TT-TN:GOTO 2100
2000 FOR I=1 TO N:IF G(I)=0 THEN 1300
2010 PS(I)=P(I):NEXT I
2015 PRINT "{up}";TAB(FN T(T));C$(G(T))
2020 H=0:FOR I=1 TO N
2030 :IF PS(I)=G(I) THEN H=H+1:PS(I)=-1:G(I)=-2
2040 NEXT I
2050 M=0:FOR I=1 TO N:FOR J=1 TO N
2060 IF PS(I)=G(J) THEN M=M+1:PS(I)=-1:G(J)=-2
2070 NEXT J:NEXT I
2080 PRINT "{up}";TAB(FN T(N+1));LEFT$(H$,H+1);LEFT$(M$,M+1)
2090 IF H<N THEN 1100
2100 PRINT "{up rvs}";TAB(FN T(1));
2110 FOR I=1 TO N-1
2120 :PRINT C$(P(I));".";:NEXT I
2130 PRINT C$(P(N))
2135 IF H=N THEN PRINT TAB(TB);"{down}...THAT WAS THE PATTERN!"
2140 PRINT "{down}CARE TO TRY AGAIN?.";:GOSUB 60000
2150 IN$=LEFT$(IN$,1):IF IN$<>"N" THEN 1000
2160 PRINT "{down}VERY WELL.{2°down}":IF LS<GM THEN 2300
2170 IF LM<40 THEN 2250
2190 PRINT ".YOU DIDN'T GUESS.";:IF GM=1 THEN 2210
2200 IF GM=2 THEN PRINT "EITHER":PRINT "OF.";:GOTO 2205
2201 PRINT "ANY OF"
2205 PRINT "THE";GM;"PATTERNS";:GOTO 2220
2210 PRINT "THE":PRINT "PATTERN";

```

```

2220 PRINT ".I.SELECTED.":GOTO 3000
2250 PRINT "TRY.A.HIGHER.GUESS.LIMIT.THE.NEXT.TIME":
PRINT "YOU.PLAY.THIS.";
2260 GOTO 2190
2300 PRINT "YOU.AVERAGED";INT(TT*10/(GM-LS))/10;
"GUESSES.PER.SOLUTION,"
2310 IF LS>0 THEN PRINT "BUT.MISSED";LS;"OUT.OF";GM;"PATTERNS.":
GOTO 3000
2320 PRINT "WITHOUT.MISSING.":IF GM=1 THEN PRINT "THE.PATTERN.":
GOTO 3000
2330 PRINT "ANY.OF.THE";GM;"PATTERNS."
3000 END
60000 IN$="":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 GOSUB 60500:PRINT "PRESS.{rvs}RETURN{off}_.TO_";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTINUE"
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN PRINT "{clr}":CR$=CHR$(13):GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;".-_.CURSOR.#";NM$;
"._(C).1981....PAGE";PN
60270 GOSUB 60500:RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;".BY.GLEN.FISHER"
62002 PRINT "{down}....CURSOR.#7..COPYRIGHT.(C).1981"
62004 GOSUB 60500
62005 PRINT "{down}GUESS{.}THE{.}COLOR.PATTERNS"
62010 PRINT "{2°down}WANT.DIRECTIONS?_";:GOSUB 60000:PRINT :
IN$=LEFT$(IN$,1)
62020 IF IN$="" OR IN$="N" THEN POKE 59468,12:PRINT "{clr}":RETURN
62030 IF IN$<>"Y" THEN PRINT "PLEASE.ANSWER.YES.OR{.}NO.{4°up}":
GOTO 62010
62100 LP=5:GOSUB 60200
62110 PRINT "MASTERMIND.IS.A.GAME.IN.WHICH.YOU.TRY"
62120 PRINT "TO.GUESS.THE.COLOR.PATTERN.THE.PET.HAS"

```



```

62130 PRINT "CHOSEN..AFTER.EACH.GUESS.YOU.MAKE,.THE"
62140 PRINT "PET.WILL.GIVE.YOU.HINTS.ABOUT.HOW.CLOSE"
62150 PRINT "YOU.CAME.{down}"
62160 PRINT "YOU.WILL.BE.TOLD.HOW.MANY.HITS.AND"
62170 PRINT "MISSES.YOU.GOT.WITH.THAT.GUESS."
62180 PRINT "{down}YOU.SCORE.A.HIT.WHENEVER.A.COLOR.IN"
62190 PRINT "YOUR.GUESS.MATCHES.A.COLOR.IN.THE.PET'S"
62200 PRINT "PATTERN.EXACTLY,.IN.COLOR.AND.POSITION."
62210 PRINT "FOR.EXAMPLE"
62220 PRINT "{down}PATTERN:..RED...GREEN...BLUE...VIOLET"
62240 PRINT "GUESS:...BLUE...GREEN...RED...ORANGE"
62250 PRINT "{down}WOULD.GET.A.HIT.FOR.THE{.}GREEN."
62260 GOSUB 60200
62270 PRINT "YOU.GET.A.MISS.WHENEVER.A.COLOR.IN.YOUR"
62280 PRINT "GUESS.MATCHES.A.COLOR.IN.THE.PATTERN,"
62290 PRINT "BUT.ISN'T.IN.THE.RIGHT.POSITION..FOR"
62300 PRINT "EXAMPLE"
62310 PRINT "{down}PATTERN:..GREEN...GREEN...BLUE....RED"
62330 PRINT "GUESS:...BLUE...ORANGE...ORANGE...VIOLET"
62340 PRINT "{down}WOULD.GET.A.MISS.ON.THE.BLUE."
62350 PRINT "{down}ANY.GIVEN.COLOR.IN.YOUR.GUESS.CAN.MATCH"
62360 PRINT "ONLY.ONE.COLOR.IN.THE.PATTERN,.EVEN.IF"
62370 PRINT "THERE.ARE.DUPLICATES..FOR.EXAMPLE"
62380 PRINT "{down}PATTERN:..RED...RED.....ORANGE...ORANGE"
62400 PRINT "GUESS:...RED...ORANGE...BLUE....GREEN"
62410 PRINT "{down}WOULD.GET.ONE.HIT.(FOR.RED).AND{.}"
62420 PRINT "ONE.MISS{.}(FOR{.}ORANGE). "
62430 GOSUB 60200
62440 PRINT "SIMILARLY,.ANY.GIVEN.COLOR{.}IN.THE"
62450 PRINT "PATTERN.CAN.MATCH.ONLY.ONE.COLOR.IN"
62460 PRINT "YOUR.GUESS..FOR.EXAMPLE"
62470 PRINT "{down}PATTERN:..RED...ORANGE...BLUE....GREEN"
62480 PRINT "GUESS:...RED...RED.....ORANGE...ORANGE"
62490 PRINT "{down}WOULD.GET.ONE.HIT{.}(FOR{.}RED),.AND"
62500 PRINT "ONE.MISS.(FOR{.}ORANGE). "
62510 PRINT "{2°down}TO{.}ENTER.YOUR.GUESS,.YOU.TYPE.IN.THE"
62520 PRINT "COLORS.YOU.WANT{.}(USING.EITHER.THE.FIRST"
62530 PRINT "LETTERS.OF.THE.WORDS{.}RED,.ORANGE,"
62540 PRINT "YELLOW,.GREEN,.BLUE,.AND{.}VIOLET,.OR.BY"
62560 PRINT "TYPING.A.DIGIT.FROM.1.TO{.}6.(THE.COLORS"
62570 PRINT "ARE.NUMBERED.IN.THE.ORDER{.}RED.THROUGH"
62580 PRINT "VIOLET)....THE.POSITION.THE{.}COLOR.WILL"
62590 PRINT "GO.INTO.IS.MARKED.WITH{. rvs}BLACK-ON-WHITE{off}."
62600 GOSUB 60200
62610 PRINT "YOU.MAY.MODIFY.YOUR.GUESS.(TO.CORRECT.A"
62620 PRINT "MISTYPE,.OR.CHANGE.YOUR.MIND).BY.USING"
62630 PRINT "THE{.}DEL,.LEFT,.RIGHT,.AND{.}SPACE{.}KEYS."
62640 PRINT "DEL.AND{.}LEFT.MOVE{.}THE{. rvs}MARKER{off}.TO.THE"
62650 PRINT "LEFT,.AND.RIGHT.AND{.}SPACE.MOVE.IT.TO"
62660 PRINT "RIGHT..IF.THE{. rvs}MARKER{off}.GOES.OFF.EITHER"
62670 PRINT "END,.IT.WILL.REAPPEAR.AT.THE.OTHER."

```

```

62680 PRINT "IN.ADDITION,,THE{.}'<'.AND{.}'>'.KEYS"
62690 PRINT "WILL.SWITCH.COLORS.. '<'.WILL.SWITCH.THE"
62700 PRINT "COLOR.UNDER.THE{. rvs}MARKER{off}.WITH.THE.COLOR"
62710 PRINT "TO.ITS.LEFT,,AND{.}'>'.WITH.THE.COLOR.TO"
62720 PRINT "THE.RIGHT.OF.THE{. rvs}MARKER{off}."
62730 PRINT "{down}ALSO,,TO.GIVE.UP.AND.SEE.THE.PATTERN,"
62740 PRINT "YOU.CAN.PRESS.THE.'?'_KEY."
62750 PRINT "{down}WHEN.YOUR.GUESS.IS.ENTERED.TO.YOUR"
62760 PRINT "SATISFACTION,,PRESS.RETURN.TO.FIND.OUT"
62770 PRINT "HOW.MANY.HITS.AND.MISSES.YOU.GOT."
62780 GOSUB 60200
62790 PRINT "THE.PET.WILL.SHOW.YOUR.HITS.AND.MISSES"
62800 PRINT "BY.PRINTING,,BESIDE.YOUR.GUESS,,A.SOLID"
62810 PRINT "DOT.FOR.EACH.HIT,,AND.A.HOLLOW.DOT.FOR"
62820 PRINT "EACH.MISS..FOR.EXAMPLE,,2.HITS.AND.1"
62830 PRINT "MISS.WOULD.LOOK.LIKE.THIS:"
62840 PRINT "{down}...{QQW}"
62860 PRINT "{down}WHEN.YOU'RE.ALL.DONE.GUESSING.PATTERNS,"
62870 PRINT "THE.PET.WILL.TELL.YOU.HOW.WELL.YOU.DID"
62880 PRINT "OVER.ALL.THE.PATTERNS."
62890 PRINT "{3°down}GOOD.LUCK.{2°down}"
63000 GOSUB 60200

```



PEG

By Julia Hallford

Here's a simple one. With two sets of four pegs, in a board with nine holes in a straight line, all you have to do is switch the two sets of pegs. Seems easy, huh? Well, there are a few rules we haven't told you yet.

The pegs are set up in the board with four to the left (marked ">") and four to the right (marked "<"), with a hole between them. On any move, you may move a peg one place left or right (according to the arrow on top); ">" pegs can move only to the right, and a "<" peg only to the left, and must move into the hole. Also you may jump over a peg going in the other direction, if you can land in the hole. These are your only choices: move one place into the hole

or jump over an opposite-way peg into the hole. Other than that, you may move any peg you like at any time.

To move, tell the PET which peg you want to move (all are numbered). Since it must move into the hole, and there's only one hole, the PET knows where its going and you don't need to tell it that. The PET won't allow any moves which violate the rules. When you've got the pegs switched, with the hole once again in the middle, you're done. The PET will tell you how many moves you took to do it (the minimum is 24). If you get stuck, or give up, move peg zero to quit.

Incidentally, the PET knows how to solve the puzzle, and will gladly show you if you ask.

```
0 CLR:POKE 59468,12:PK=12
6 REM PEG COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/6/81 RJ
90 PG$="PEG":NM$="11":GOSUB 62000
100 GOSUB 60400
105 BB$="....."
110 PRINT "{down}WANT_TO_SEE_HOW_TO_SOLVE_THE_PUZZLE? ";
120 GOSUB 60000
130 PRINT "{clr}";:GOSUB 1000
140 IF LEFT$(IN$,1)="Y" THEN 1200
150 M=0:X=5
160 PRINT "{home}";BB$
180 M=M+1:DL=0:PRINT "{home}MOVE.#"M:Y=X
190 GOSUB 800:GOSUB 700:DL=1000
200 PRINT TAB(6);"FROM? ";:GOSUB 60000
201 IF IN$="" THEN PRINT "{up}";:GOTO 200
202 IF ASC(IN$)=32 THEN IN$=MID$(IN$,2):GOTO 201
205 X=VAL(IN$)
210 IF X=0 THEN 470
220 IF X<1 OR X>9 THEN PRINT "TYPE_A_NUMBER_FROM_1_TO_9,_PLEASE.":
GOTO 190
230 IF ABS(X-Y)>2 THEN 450
240 IF P(X)=32 GOTO 450
260 IF P(X)=62 AND X>Y THEN 450
270 IF P(X)=60 AND X<Y THEN 450
280 P(Y)=P(X):P(X)=32
290 JX=CRT+WD*8+9+2*X:JY=CRT+WD*8+9+2*Y
320 POKE JY,P(Y):POKE JX,32
```

[illegible]

```

1620 GOSUB 1000:GOTO 150
2000 DATA 62,62,62,62,32,60,60,60,60
2030 DATA 6,5,4,6,3,4,5,3,7,5,8,7,6,8,4,6,2,4,1,2,3,1,5,3
2040 DATA 7,5,9,7,8,9,6,8,4,6,2,4,3,2,5,3,7,5,6,7,4,6,5,4
60000 IN$=" ":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 GOSUB 60500:PRINT "PRESS_{rvs}RETURN{off}_{TO}_";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN POKE 59468,PK:CLR:PRINT "{clr}";:GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_-_{CURSOR}_#";NM$;
      "_{(C)}_1981.....PAGE";PN
60270 GOSUB 60500:RETURN
60400 CRT=32768:CR$=CHR$(13):WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60420 RETURN
60500 FOR I=1 TO 13:PRINT "@@@";:NEXT:PRINT :RETURN
62000 PRINT "{clr down}";TAB(15-LEN(PG$));PG$;"_BY_JULIA_HALLFORD"
62010 PRINT "{down}.....CURSOR_#11..COPYRIGHT_(C)_1981"
62020 GOSUB 60500
62025 PRINT "{down}REVERSE_TWO_SETS_OF_PEGS_ON_A_BOARD"
62030 PRINT "{2°down}WANT_INSTRUCTIONS?_";:GOSUB 60000
62040 IF LEFT$(IN$,1)<>"Y" THEN PRINT "{clr}";:POKE 59468,PK:CLR:
      GOTO 100
62100 LP=2:GOSUB 60200
62400 PRINT "THIS_GAME_IS_PLAYED_WITH_TWO_SETS_OF"
62410 PRINT "{down}FOUR_PEGS_REPRESENTED_BY_'<{'right}AND'>'."
62420 PRINT "{down}THE_BOARD_STARTS_OUT_WITH_A_SET_OF_PEGS"
62430 PRINT "{down}AT_EACH_{ }END,_WITH_A_SPACE_BETWEEN_THEM."
62440 PRINT "{2°down}THE_OBJECT_OF_THE_GAME_IS_TO_SWITCH_THE"
62450 PRINT "{down}TWO_SETS_OF_{ }PEGS..A_PEG_MAY_EITHER_MOVE"
62460 PRINT "{down}INTO_THE_SPACE,_IF_THE_SPACE_IS_NEXT_TO"
62470 PRINT "{down}THE_PEG,_OR_JUMP_OVER_ONE_PEG_INTO_THE"
62480 PRINT "{down}SPACE.._'>'_PEGS_MOVE_TO_THE_RIGHT,_AND"

```

```
62490 PRINT "{down}'<'_PEGS_TO_THE_LEFT."
62495 GOSUB 60200
62500 PRINT "TO_MOVE,_TYPE_THE_NUMBER_OF_THE_SPACE"
62510 PRINT "{down}FROM_WHICH_YOU_WANT_THE_PEG_MOVED,_AND"
62520 PRINT "{down}HIT_{rvs}RETURN{off}._IF_YOU_REACH_A_POSITION"
62530 PRINT "{down}IN_WHICH_YOU_HAVE_NO_LEGAL_MOVE,_ENTER"
62545 PRINT "{down}A_'0'_TO_RESIGN."
62550 PRINT "{2°down}THE_PUZZLE_CAN_BE_SOLVED_IN{ }24_MOVES."
62555 PRINT "{down}SEE_HOW_WELL_YOU_CAN_DO."
62560 GOSUB 60200
```



By Glen Fisher

You are playing against yourself in LEAP, trying to eliminate all but one of the pegs on the board. When the game starts, you have a 4×8 board filled with holes which contain 31 pegs. The rows are numbered and the columns are lettered, so any given hole is described by a letter/number combination (for example, D2 or 2D). The PET will ask you where you want the space to be. You may either name the place for the hole, or just press [RETURN] and let the PET pick a spot at random.

On each move, you must remove a peg by leaping over it with another. Pegs can only leap into holes. Each peg that you jump over is removed, and a hole is left in its place. The *only* kind of move you can make is a leap; just moving the peg is illegal. The game is over when you can't make any more leaps. To achieve a perfect game, you must leave only one peg behind. (Two isn't hard, but one is very difficult.)

To leap over a peg, type the positions of the peg you want to move when the PET gives you the “From ?” prompt. Then type the position of the hole you want to leap into when the PET gives you the “To ?” prompt. For the “To ?” prompt, you can give just the part of the position that changes. (For example, with a move from C4 to

C2, you could respond to “To ?” with just a “2”, since the “C” stays the same.) If you decide you want to move another peg, answer the “To ?” prompt with a [left arrow], and you’ll get the “From ?” prompt back again. You can move any direction except diagonally. If you want to resign, press [R].

PAGE 1

	A	B	C	D	E	F	G	H
1	●	●	●	●	●	●	●	●
2	■	●	●	●	●	●	●	●
3	●	●	●	●	●	●	●	●
4	□	●	●	●	●	●	●	●

TOT 2A

[illegible]

```

1050 PRINT TAB(HT); "{-@!@!@!@!@!@!@!@!}"
1110 FOR I=1 TO 4:FOR J=1 TO 8:B(I,J)=1:NEXT J:NEXT I
1200 FOR C=1 TO 8:FOR R=1 TO 4
1210 GOSUB 9200
1220 NEXT R,C
1300 IF HC>0 THEN R=HR:C=HC:GOTO 1330
1310 H=-1:V=9:GOSUB 9000:PRINT "WHERE'S THE HOLE?...{2°left}";:
    GOSUB 60000:GOSUB 9100
1315 IF IN$="" THEN R=INT(RND(1)*3)+1:C=INT(RND(1)*8)+1:ER=0
1320 IF ER THEN 1310
1325 HR=R:HC=C
1330 B(R,C)=0:GOSUB 9200
1340 H=-1:V=9:GOSUB 9000:PRINT "....."
1900 MV=0:TIS="000000"
2000 H=5:V=-3:GOSUB 9000:PRINT "{rvs}MOVE";MV+1:FR=0:FC=0
2005 H=0:V=9:GOSUB 9000:PRINT "FROM?...{2°left}";:GOSUB 60000:
    GOSUB 9100
2010 ON ER+1 GOTO 2020,2000,2950,2800
2020 FR=R:FC=C
2030 IF B(FR,FC)=0 THEN 2000
2040 GOSUB 9250
2050 H=0:V=9:GOSUB 9000:PRINT "TO?.....{4°left}";:GOSUB 60000:
    GOSUB 9100
2055 ON ER+1 GOTO 2060,2050,2950,2800
2060 TR=R:TC=C
2070 IF B(TR,TC)<>0 THEN 2050
2100 IF FR<>TR AND FC<>TC THEN 2900
2110 IF ABS(FR-TR)<>2 AND FC=TC THEN 2900
2120 IF ABS(FC-TC)<>2 AND FR=TR THEN 2900
2130 MR=(FR+TR)/2:MC=(FC+TC)/2
2140 IF B(MR,MC)=0 THEN 2900
2200 B(TR,TC)=B(FR,FC):B(FR,FC)=0
2210 B(MR,MC)=0
2220 R=FR:C=FC:GOSUB 9200
2230 R=TR:C=TC:GOSUB 9200
2240 R=MR:C=MC:GOSUB 9200
2250 MV=MV+1
2260 H=2:V=13:GOSUB 9000:PRINT "TIME.";ET$=TIS
2270 PRINT LEFT$(ET$,2);": ";MID$(ET$,3,2);": ";RIGHT$(ET$,2)
2300 FOR I=1 TO 4:FOR J=1 TO 8:T=SGN(B(I,J))
2310 IF I<3 THEN IF T<>SGN(B(I+2,J)) AND B(I+1,J)<>0 THEN 2000
2320 IF J<7 THEN IF T<>SGN(B(I,J+2)) AND B(I,J+1)<>0 THEN 2000
2330 NEXT J,I
2400 H=0:V=11:GOSUB 9000:PRINT "YOU HAVE NO MOVE."
2410 PG=31-MV:IF PG<BP THEN BP=PG:BT$=ET$
2420 IF PG=BP AND ET$<BT$ THEN BT$=ET$
2500 H=0:V=15:GOSUB 9000:PRINT "TRY AGAIN?.";:GOSUB 60000
2510 IF LEFT$(IN$,1)<>"N" THEN 1000
2520 PRINT "{clr 2°down}YOUR BEST GAME WAS:.....{down}"
2530 PRINT BP;"PEGS LEFT IN.";
2540 IF LEFT$(ET$,2)<>"00" THEN PRINT LEFT$(ET$,2);": ";

```



```

2550 PRINT MID$(BT$,3,2);": ";RIGHT$(BT$,2);"."
2590 END
2800 ET$=TI$:GOTO 2410
2900 IF ER<0 THEN 2950
2905 H=4:V=11:GOSUB 9000:PRINT "{rvs}BAD_MOVE."
2910 TM=TI+90
2920 IF TI<TM THEN 2920
2930 GOSUB 9000:PRINT "....."
2950 R=FR:C=FC:GOSUB 9200
2960 GOTO 2000
8999 END
9000 PRINT "{home}":POKE QL,V+VT:PRINT "{up}";TAB(HT+H);:RETURN
9100 ER=1:IF IN$="<" THEN ER=2
9101 IF IN$="R" THEN ER=3
9102 IF LEN(IN$)=1 AND FR<>0 THEN 9170
9105 IF LEN(IN$)<>2 THEN RETURN
9110 R$=LEFT$(IN$,1):C$=RIGHT$(IN$,1)
9120 IF R$>="A" AND R$<="H" THEN 9150
9130 IF R$<"1" OR R$>"4" OR C$<"A" OR C$>"H" THEN RETURN
9140 R=ASC(R$)-48:C=ASC(C$)-64:ER=0:RETURN
9150 IF C$<"1" OR C$>"4" THEN RETURN
9160 R=ASC(C$)-48:C=ASC(R$)-64:ER=0:RETURN
9170 IF IN$>="A" AND IN$<="H" THEN C=ASC(IN$)-64:R=FR:ER=0:RETURN
9180 IF IN$>="1" AND IN$<="4" THEN R=ASC(IN$)-48:C=FC:ER=0:RETURN
9190 RETURN
9200 H=2*C-1:V=2*R-1:GOSUB 9000:PRINT MID$(DT$,1+2*B(R,C),2):RETURN
9250 B=B(R,C):B(R,C)=B+3:GOSUB 9200:B(R,C)=B:RETURN
9999 END
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>2 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{TO}_";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240

```

```

60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;"_._CURSOR.#";NM$;
      "_._(C)._1981._._.PAGE";PN
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 QL=245:CR$=CHR$(13)
60410 IF PEEK(50000)=0 THEN RETURN
60420 QL=216:POKE 1022,128
60430 RETURN
60500 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;"_._BY_GLEN_FISHER"
62010 PRINT "{down}._._._CURSOR.#13._._COPYRIGHT._(C)._1981"
62020 GOSUB 60500
62030 PRINT "{down}CLEAR_THE_BOARD_OF_ALL_THE_PEGS"
62080 PRINT "{2°down}WANT_DIRECTIONS?_";:GOSUB 60000
62090 IF LEFT$(IN$,1)<>"Y" THEN 60300
62100 LP=2:GOSUB 60200
62110 PRINT "YOU_BEGIN_WITH_A_4_BY_8_BOARD_WITH_ALL"
62120 PRINT "{down}SQUARES_FILLED_WITH_WHITE_PEGS._YOU"
62130 PRINT "{down}FIRST_REMOVE_ONE_PEG._(IF_YOU_PRESS"
62140 PRINT "{down rvs}RETURN{off},_A_RANDOM_PEG_IS_REMOVED.)"
62150 PRINT "{2°down}AFTER_THE_STARTING_'HOLE'_IS_SELECTED,"
62160 PRINT "{down}THE_OBJECT_OF_THE_GAME_IS_TO_REMOVE_PES{left
      ins}G"
62170 PRINT "{down}BY_JUMPING_OTHER_PEGS_OVER_THEM._NOTE"
62180 PRINT "{down}THAT_YOU_MAY_ONLY_JUMP_UP_AND_DOWN,_OR"
62190 PRINT "{down}LEFT_AND_RIGHT,_BUT_{rvs}NOT{off}_DIAGONALLY!"
62195 GOSUB 60200
62200 PRINT "THE_GAME_ENDS_WHEN_YOU_HAVE_ONLY_ONE_PG{left ins}E"
62210 PRINT "{down}LEFT,_OR_WHEN_YOU_HAVE_NO_REMAINING"
62220 PRINT "{down}LEGAL_MOVES._YOU_CAN_'RESIGN'_BY_PRESS-"
62230 PRINT "{down}ING_'R'_FOR_YOUR_'FROM'_MOVE."
62240 PRINT "{2°down}IF_YOU_CHANGE_YOUR_MIND_ABOUT_WHICH_PEG"
62250 PRINT "{down}YOU_WANT_TO_MOVE,_ENTER_A_'<'_FOR_YOUR"
62260 PRINT "{down}'TO'_MOVE,_AND_YOU'LL_BE_ASKED_AGAIN"
62270 PRINT "{down}WHICH_PEG_YOU_WISH_TO_MOVE."
62280 GOSUB 60200

```



By David Plotton

This is our version of the ancient “Fifteen” puzzle. You’re given a board with 16 spaces, 15 of which have tiles numbered 1 through 15. By sliding the tiles into the empty square, you must move the numbers around so that they read from 1 to 15 in order. Naturally, the fewer moves it takes you, the better.

After the board is displayed to you, start sliding tiles around. To slide a tile, use the numeric pad to tell which direction the tile should move. Since there is only one hole, there is only one tile that can slide in that direction. Below the board is a counter which keeps track of each move. You may be surprised at how many moves it takes you.

The PET will watch you until you get the tiles in order, one to four in the top row, five to eight in the second, and so on, with the hole in the lower right corner.

2 MOVES

2 MOVES

```

0 CLR:POKE 59468,12
6 REM FIFTEEN COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/5/81 TZ
90 PG$="FIFTEEN":NM$="15":GOSUB 62000
100 DIM A(16),B(16):TB=8
101 PRINT CHR$(142)
120 PRINT "{clr 2°down}"TAB(TB){0@@@@2@@@@2@@@@2@@@@.}"
130 FOR I=1 TO 4:FOR J=1 TO 3:PRINT TAB(TB);
    "{}}.....{}}.....{}}.....{}}.....{}}":NEXT J
140 IF I<4 THEN PRINT TAB(TB);"{+@@@@[@@@@[@@@@[@@@@3}"
150 NEXT I:PRINT TAB(TB);"{-@@@@1@@@@1@@@@1@@@@=}"
160 FOR I=1 TO 16:A(I)=I:NEXT I
170 FOR I=15 TO 2 STEP -1:J=RND(1)*I+1:T=A(J):A(J)=A(I):A(I)=T:
    NEXT I
180 FOR I=1 TO 16:B(I)=A(I):IF B(I)=16 THEN BS=I
190 NEXT I
200 C=0:FOR I=1 TO 16
210 IF A(I)<>I THEN T=A(I):A(I)=A(T):A(T)=T:C=C+1:GOTO 210
220 NEXT I
230 IF (C AND 1)=0 THEN 270
240 I=INT(RND(1)*16)+1:IF B(I)=16 THEN 240
250 J=INT(RND(1)*16)+1:IF B(J)=16 OR I=J THEN 240
260 T=B(I):B(I)=B(J):B(J)=T
270 PRINT "{home}":PRINT "{3°down}";
280 FOR I=1 TO 16 STEP 4:FOR J=0 TO 3:PRINT TAB(TB+J*5+2);
290 IF B(I+J)<16 THEN PRINT RIGHT$(STR$(B(I+J)),2);

```

```

300 NEXT J:N=0:PRINT "{3°down}":NEXT I:POKE QL,22:
    PRINT "{up}"TAB(13)N;"MOVES"
310 GET Z$:IF Z$="" THEN 310
315 IF Z$="Q" THEN PRINT "{clr}":END
320 IF Z$<"2" OR Z$>"8" THEN 310
330 IF VAL(Z$) AND 1 THEN 310
340 ON VAL(Z$)/2 GOTO 410,390,370,350
350 T=BS+4:IF T>16 THEN 310
360 GOTO 420
370 T=BS-1:IF (3 AND T)=0 THEN 310
380 GOTO 420
390 T=BS+1:IF (3 AND BS)=0 THEN 310
400 GOTO 420
410 T=BS-4:IF T<1 THEN 310
420 B(BS)=B(T):B(T)=16
430 I=INT((BS-1)/4):J=BS-I*4
440 POKE QL,I*4+5:PRINT "{up}"TAB(TB+J*5-3)RIGHT$(STR$(B(BS)),2)
450 I=INT((T-1)/4):J=T-I*4
460 POKE QL,I*4+5:PRINT "{up}"TAB(TB+J*5-3)"_._"
470 BS=T:N=N+1:POKE QL,22:PRINT "{up}"TAB(13)STR$(N);"_MOVE";:
    IF N>1 THEN PRINT "S";
475 PRINT "_":IF BS<16 THEN 310
480 FOR I=1 TO 16:IF B(I)<>I THEN 310
490 NEXT I:POKE QL,21:PRINT "{up}"TAB(14)"SOLVED_ IN{2°down}"
600 PRINT "DO_YOU_WANT_TO_PLAY_AGAIN?_";:GOSUB 60000:IF LEFT$(IN$,
    1)<>"N" THEN 120
700 END
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_&{&}",ZC,1);"{left}";:ZC=3-ZC:
    ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
    GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
    Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
    GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
    NEXT Z:GOTO 60000
60140 GOTO 60010
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 QL=245:CR$=CHR$(13)
60410 IF PEEK(50000) THEN QL=216
60430 RETURN
60500 FOR I=1 TO 13:PRINT "@@@";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;"_BY_DAVID_PLATTON"
62010 PRINT "{down}_.CURSOR_#15_.COPYRIGHT_(C)_1981"

```

```
62020 GOSUB 60500
62030 PRINT "{down}THE.FIFTEEN.PUZZLE"
62040 PRINT "{2°down}PRESS.{rvs}RETURN{off}.TO.BEGIN"
62050 GET T$:IF T$="" THEN 62050
62060 GOTO 60300
```



RATRUN

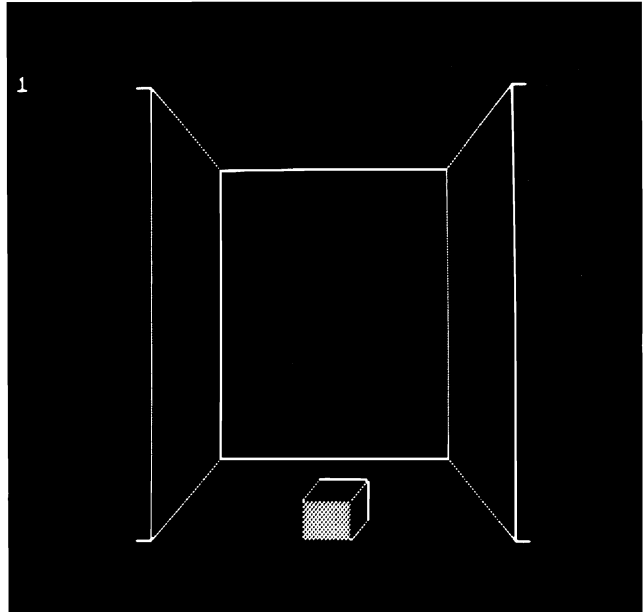
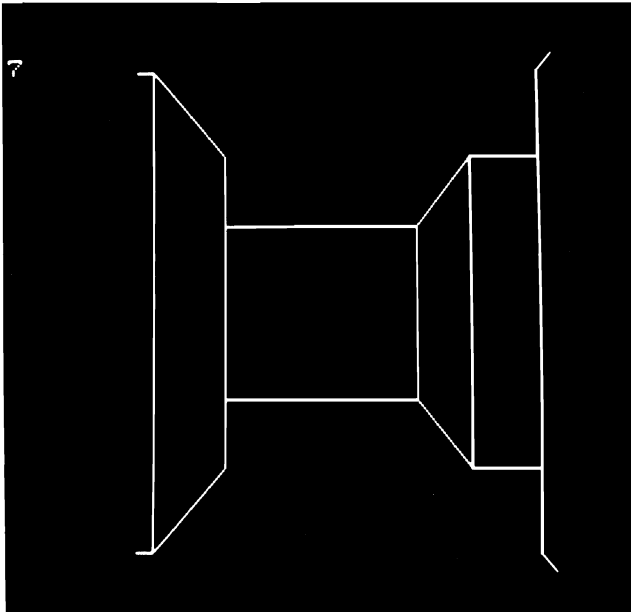
By Chris Nadovich

Yes indeed, here is yet another maze-making program. However, this one is different. In this program, you are *inside* the maze, not above it. The goal is a piece of cheese (this is a *rat* run, after all) somewhere in the maze. Since the rat can probably smell the cheese better than you can, we'll give you one additional clue. In a corner of the screen, you're told how far it is to the cheese. That's not as much help as you might think; the quoted distance is "as the flea jumps," ignoring the walls of the maze. Thus, it tells how far away you are from the cheese, not how far you must travel to reach it.

To navigate the maze, you may press the [8] key to move forward. You may also press the [4]

key to turn to the left, [6] to turn to the right, or [2] to turn around backward. The [4], [6], and [2] keys do *not* move you, they just turn you. Only the [8] key moves you.

Finally, if you despair of ever finding the cheese, press [?]. You'll be given a map of the maze. Where you started will be marked with a big white square, your travels will be shown by periods in the squares you've occupied, and the cheese will be marked with a grey square. Do not give up easily. Once you see the map, you'll never be allowed into *that* maze again. Other mazes are fine, (and you'll be offered one), but never that one. **WARNING:** If you get nervous in elevators or closed spaces, do not play this game alone at night.

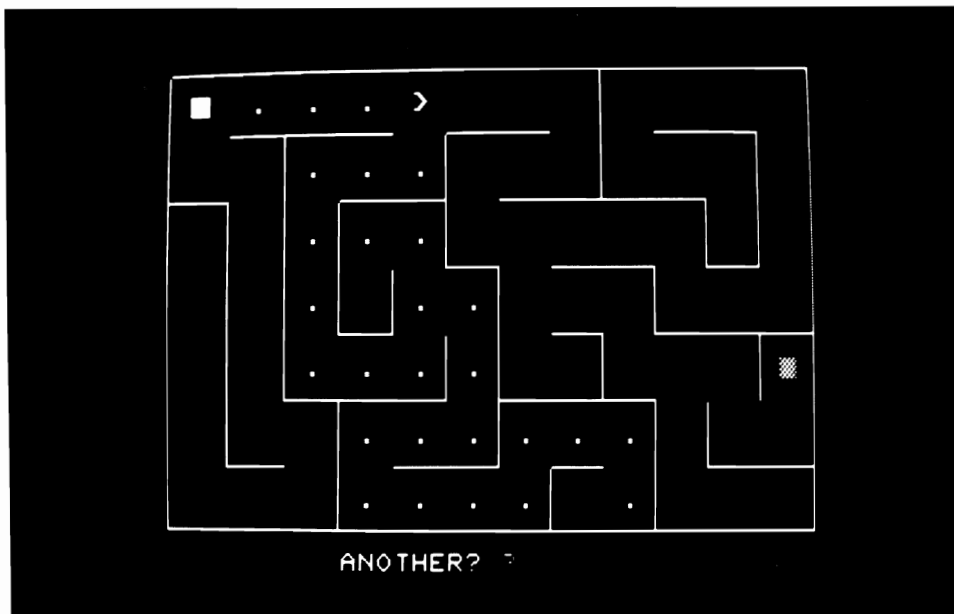


```
0 CLR:POKE 59468,12
6 REM  RATRUN COPYRIGHT (C) 1981 THE CODE WORKS
10 REM  AS OF 10/8/81 RJ
90 PG$="RATRUN":NM$="13":GOSUB 62000
100 PRINT CHR$(142):H=12:V=7:H1=H+1:W=634:V1=826
110 I=RND(-TI)
130 DIM XL(4),N2(4),YL(4),XR(4)
140 FOR I=0 TO 4:READ XL(I),N2(I),YL(I),XR(I):NEXT I
150 DATA 8,20,-4,31,8,12,1,27,12,6,5,24,15,2,8,22,17,0,10,21
200 CX=INT(RND(1)*H)+1:CY=INT(RND(1)*V)+1
210 C=CX:R=CY:S=0
```

```

220 FOR I=0 TO (V+1)*H1:POKE W+I,0:POKE V1+I,0:NEXT
230 PRINT "{clr 2°down}SETTING UP MAZE...";
300 DA=0:N=0:AV=V1+C+R*H1:AW=W+C+R*H1:POKE AW,1:S=S+1:
    IF S>=H*V THEN 900
320 PRINT MID$("{rvs off}",(S AND 1)+1,1);"{V left}";
410 IF C>1 AND PEEK(AW-1)=0 THEN DA=DA+2:N=N+1
420 IF C<H AND PEEK(AW+1)=0 THEN DA=DA+1:N=N+1
430 IF R>1 AND PEEK(AW-H1)=0 THEN DA=DA+8:N=N+1
440 IF (R<V AND PEEK(AW+H1)=0) THEN DA=DA+4:N=N+1
450 N=INT(RND(1)*N)+1
460 ON DA+1 GOTO 500,850,800,600,750,610,620,630,700,640,650,660,
    670,680,690,695
500 S=S-1
510 C=C+1:IF C>H THEN C=1:R=R+1:IF R>V THEN R=1
520 ON PEEK(W+C+R*H1)+1 GOTO 510,300
600 ON N GOTO 800,850
610 ON N GOTO 750,850
620 ON N GOTO 750,800
630 ON N GOTO 750,800,850
640 ON N GOTO 700,850
650 ON N GOTO 700,800
660 ON N GOTO 700,800,850

```



```

670 ON N GOTO 700,750
680 ON N GOTO 700,750,850
690 ON N GOTO 700,750,800
695 ON N GOTO 700,750,800,850
700 R=R-1:AV=AV-H1:POKE AV,PEEK(AV) OR 1:GOTO 300
750 POKE AV,PEEK(AV) OR 1:R=R+1:AV=AV+H1:GOTO 300
800 C=C-1:AV=AV-1:POKE AV,PEEK(AV) OR 2:GOTO 300
850 POKE AV,PEEK(AV) OR 2:C=C+1:AV=AV+1:GOTO 300
900 I=RND(1)>.5:J=RND(1):K=RND(1)>.5
920 IF I THEN X=INT(J*H)+1:Y=V+(V-1)*K

```

```

930 IF I=0 THEN Y=INT(J*V)+1:X=H+(H-1)*K
931 IF (CX-X)2+(CY-Y)2<(H2+V2)/9 THEN 900
950 SX=X:SY=Y
1001 PRINT "{clr}"
1010 FOR I=W TO W+(V+1)*H1:POKE I,0:NEXT
1020 D=INT(RND(1)*4)
1030 T4=TI:POKE (W+X+Y*H1),1
1045 IF F=0 THEN GOSUB 3000:PRINT "{home}";
      INT(SQR((X-CX)2+(Y-CY)2)*10+.5)/10
1049 IF CX=X AND CY=Y THEN 6000
1050 GET Q$:IF Q$=""GOTO 1050
1060 IF Q$="Q" OR Q$="?" THEN K=TI:GOTO 6100
1070 I=VAL(Q$):IF I=0 THEN 1050
1075 IF I=2 THEN D=D+2
1080 IF I=4 THEN D=D-1
1090 IF I=6 THEN D=D+1
1100 IF D<0 THEN D=D+4
1110 IF D>3 THEN D=D-4
1120 IF I<>8 THEN 1045
1200 AV=V1+X+Y*H1:AW=W+X+Y*H1
1201 ON D GOTO 1260,1310,1360
1210 IF Y>1 AND (PEEK(AV-H1) AND 1) THEN Y=Y-1:POKE AW-H1,1:
      GOTO 1045
1220 GOTO 1400
1260 IF X<H AND (PEEK(AV) AND 2) THEN X=X+1:POKE AW+1,1:GOTO 1045
1290 GOTO 1400
1310 IF Y<V AND (PEEK(AV) AND 1) THEN Y=Y+1:POKE AW+H1,1:GOTO 1045
1340 GOTO 1400
1360 IF X>1 AND (PEEK(AV-1) AND 2) THEN X=X-1:POKE AW-1,1:GOTO 1045
1400 DN=23:GOSUB 7000
1410 PRINT TAB(16);"{$$$$$$$ 7°left down rvs}NO_DOOR{up}":FOR I=1 TO
      1000:NEXT
1420 PRINT TAB(16);"{up}.....{7°left down}.....{up}":GOTO 1049
2000 PRINT "{clr}.";
2020 FOR I=1 TO H:PRINT "$$$";:NEXT:PRINT "{2°down}"
2100 FOR J=1 TO V:PRINT "' up left ' up left ' 2°down";:FOR I=1 TO
      H
2125 K=PEEK(V1+I+J*H1)
2130 PRINT MID$("{$:}...{'$$$}...",1+3*K,3);
2140 IF K<2 THEN PRINT "{up left ' up left ' 2°down}";
2200 PRINT "{up left}";
2210 IF PEEK(W+I+J*H1)=1 THEN PRINT "{left}.";
2250 IF I=SX AND J=SY THEN PRINT "{left rvs}._{off}";
2260 IF I=CX AND J=CY THEN PRINT "{left &}";
2270 IF I=X AND J=Y THEN PRINT "{left}";MID$("↑>V<",D+1,1);
2280 PRINT "{right down}";
2290 NEXT:PRINT "{2°down}":NEXT:RETURN
3000 S=0:PRINT "{clr}"
3005 AV=V1+X+Y*H1
3010 ON D GOTO 3200,3400,3600
3025 IF SY=Y-S AND SX=X THEN GOSUB 5100

```



```

3026 IF CY=Y-S AND CX=X THEN GOSUB 5000
3027 K=PEEK(AV-(S+1)*H1):IF K=0 OR K=2 GOTO 3100
3030 IF PEEK(AV-S*H1) AND 2 THEN GOSUB 4000:GOTO 3040
3035 GOSUB 4200
3040 IF PEEK(AV-1-S*H1) AND 2 THEN GOSUB 4500:GOTO 3050
3045 GOSUB 4700
3050 S=S+1:IF S>4 GOTO 3900
3060 GOTO 3025
3100 IF PEEK(AV-S*H1) AND 2 THEN GOSUB 4300:GOTO 3110
3105 GOSUB 4800
3110 IF PEEK(AV-1-S*H1) AND 2 THEN GOSUB 4400:GOTO 3120
3115 GOSUB 4900
3120 RETURN
3200 IF CY=Y AND CX=X+S THEN GOSUB 5000
3210 IF SY=Y AND SX=X+S THEN GOSUB 5100
3226 K=PEEK(AV+S):IF K=0 OR K=1 GOTO 3300
3230 IF PEEK(AV+S)=2 THEN GOSUB 4200:GOTO 3240
3235 GOSUB 4000
3240 IF PEEK(AV+S-H1) AND 1 THEN GOSUB 4500:GOTO 3250
3245 GOSUB 4700
3250 S=S+1:IF S>4 GOTO 3900
3260 GOTO 3200
3300 IF PEEK(AV+S)=0 THEN GOSUB 4800:GOTO 3310
3305 GOSUB 4300
3310 IF PEEK(AV+S-H1) AND 1 THEN GOSUB 4400:GOTO 3320
3315 GOSUB 4900
3320 RETURN
3400 IF CY=Y+S AND CX=X THEN GOSUB 5000
3410 IF SY=Y+S AND SX=X THEN GOSUB 5100
3426 K=PEEK(AV+S*H1):IF K=0 OR K=2 THEN GOTO 3500
3430 IF PEEK(AV-1+S*H1) AND 2 THEN GOSUB 4000:GOTO 3440
3435 GOSUB 4200
3440 IF PEEK(AV+S*H1)=1 THEN GOSUB 4700:GOTO 3450
3445 GOSUB 4500
3450 S=S+1:IF S>4 GOTO 3900
3460 GOTO 3400
3500 IF PEEK(AV-1+S*H1) AND 2 THEN GOSUB 4300:GOTO 3510
3505 GOSUB 4800
3510 IF PEEK(AV+S*H1)=0 THEN GOSUB 4900:GOTO 3520
3515 GOSUB 4400
3520 RETURN
3600 IF CX=X-S AND CY=Y THEN GOSUB 5000
3610 IF SX=X-S AND SY=Y THEN GOSUB 5100
3625 K=PEEK(AV-(S+1)):IF K=0 OR K=1 THEN GOTO 3700
3630 IF PEEK(AV-S-H1) AND 1 THEN GOSUB 4000:GOTO 3640
3635 GOSUB 4200
3640 IF PEEK(AV-S) AND 1 THEN GOSUB 4500:GOTO 3650
3645 GOSUB 4700
3650 S=S+1:IF S>4 THEN 3900
3660 GOTO 3600
3700 IF PEEK(AV-S-H1) AND 1 THEN GOSUB 4300:GOTO 3710

```

```

3705 GOSUB 4800
3710 IF PEEK(AV-S) AND 1 THEN GOSUB 4400:GOTO 3720
3715 GOSUB 4900
3720 RETURN
3900 DN=11:GOSUB 7000:PRINT TAB(19){MN down 2°left NM home}:RETURN
4000 DN=YL(S):GOSUB 7000
4015 IF S=0 THEN PRINT TAB(XR(S));"${}$"
4020 IF S>0 AND S<4 THEN FOR I=1 TO 4-S:PRINT TAB(XR(S)+5-S){%}:
    NEXT
4045 IF S>0 THEN PRINT TAB(XR(S))MID$("${$$$%}",S)
4070 IF S=4 THEN PRINT TAB(XR(S)){%% down 2°left %%}
4071 Q$="":IF S>0 THEN Q$=MID$("{3°right %}",S)
4072 IF S<4 THEN FOR I=0 TO N2(S)+1:PRINT TAB(XR(S)){%}"Q$:NEXT
4075 IF S>0 THEN PRINT TAB(XR(S))MID$("#{###%}",S)
4095 IF S>0 AND S<4 THEN FOR I=1 TO 4-S:PRINT TAB(XR(S)+5-S){%}:
    NEXT
4096 IF S=0 THEN PRINT TAB(XR(S)){#}
4098 PRINT "{home}"
4099 RETURN
4200 DN=YL(S):GOSUB 7000
4215 IF S=0 THEN PRINT TAB(XR(S)){N}:GOTO 4230
4220 FOR I=1 TO 5-S:PRINT TAB(XR(S)+5-S-I){N}:NEXT
4230 DN=N2(S)+2:GOSUB 7100
4235 IF S=0 THEN PRINT TAB(XR(S)){M}:GOTO 4249
4240 FOR I=0 TO 4-S:PRINT TAB(XR(S)+I){M}:NEXT
4249 PRINT "{home}"
4250 RETURN
4300 J=3-S:IF S=0 THEN J=0
4305 PRINT "{home}";
4306 IF S>0 GOTO 4328
4307 PRINT TAB(20);:PRINT "{down}";:FOR I=1 TO 18:PRINT "{#}";:NEXT:
    PRINT "{#}"
4310 DN=N2(0):GOSUB 7100
4320 PRINT TAB(20);:FOR I=1 TO 19:PRINT "${}";:NEXT:GOTO 4398
4328 DN=YL(S):GOSUB 7100
4329 FOR I=1 TO 5-S:PRINT TAB(J+XR(S)+2){%}:NEXT
4330 PRINT TAB(20);:FOR I=19 TO J+XR(S):PRINT "{#}";:NEXT:
    PRINT "{%}"
4370 IF S<4 THEN FOR I=1 TO N2(S):PRINT TAB(J+XR(S)+2){%}:NEXT
4380 PRINT TAB(20);:FOR I=19 TO J+XR(S):PRINT "${}";:NEXT:
    PRINT "{%}"
4385 FOR I=1 TO 5-S:PRINT TAB(J+XR(S)+2){%}:NEXT
4398 PRINT "{home}"
4399 RETURN
4400 PRINT "{home}";
4406 IF S>0 GOTO 4430
4407 PRINT "{down}";:FOR I=1 TO 11+XL(0):PRINT "{#}";:NEXT:
    PRINT "{#}"
4410 DN=N2(0):GOSUB 7100
4420 FOR I=1 TO 12+XL(0):PRINT "${}";:NEXT:GOTO 4398
4430 DN=YL(S):GOSUB 7100

```

```

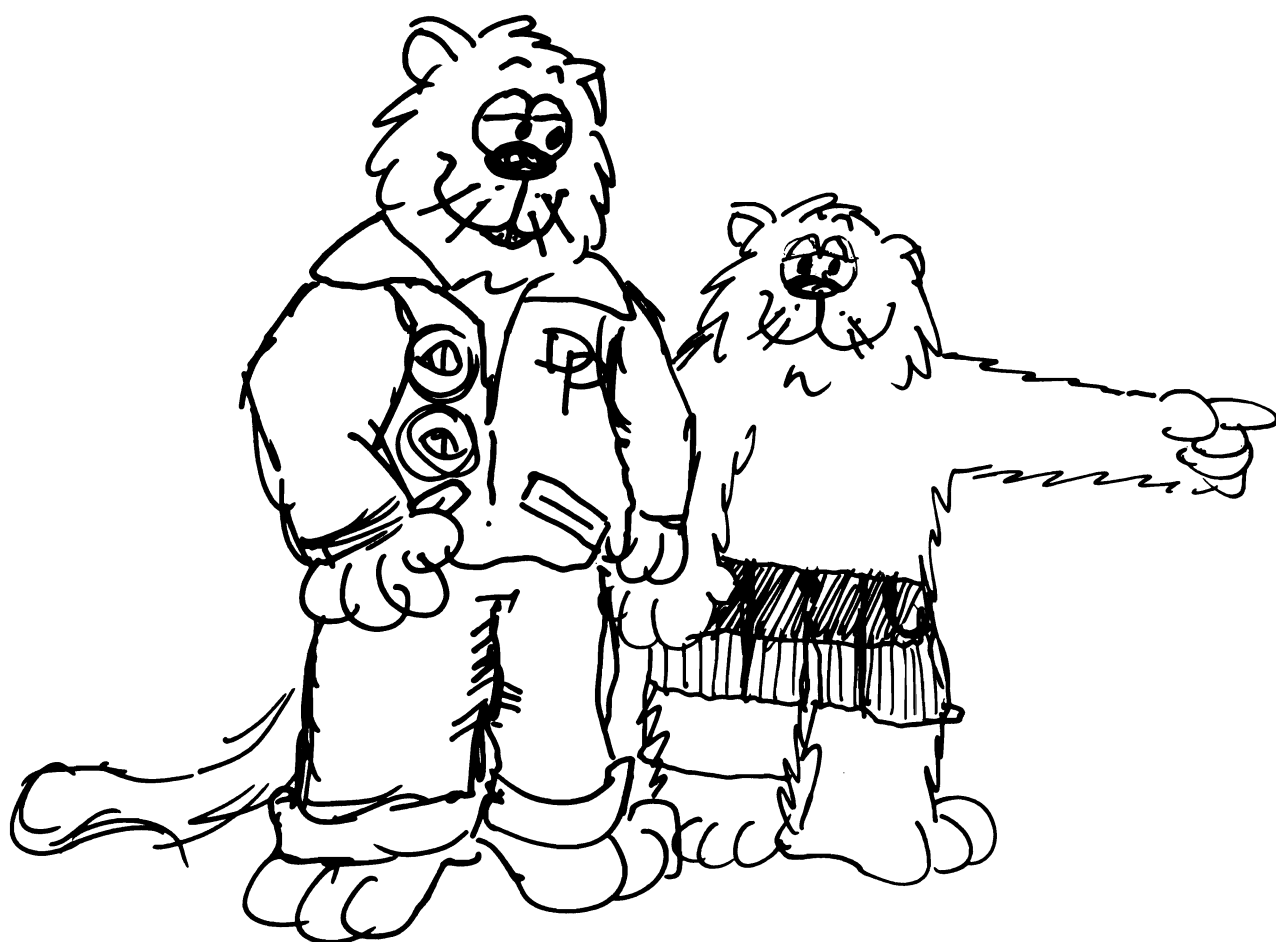
4431 FOR I=1 TO 5-S:PRINT TAB(XL(S))"{'}":NEXT
4432 PRINT TAB(XL(S));:PRINT "{#}";:FOR I=XL(S) TO 17:PRINT "{#}";:
NEXT:PRINT "{#}"
4470 IF S<4 THEN FOR I=1 TO N2(S):PRINT TAB(XL(S))"{'}":NEXT
4480 PRINT TAB(XL(S));:PRINT "{'}";:FOR I=XL(S) TO 17:PRINT "{$}";:
NEXT:PRINT "{$}"
4485 FOR I=1 TO 5-S:PRINT TAB(XL(S))"{'}":NEXT
4498 PRINT "{home}"
4499 RETURN
4500 DN=YL(S):GOSUB 7000
4515 IF S=0 THEN PRINT TAB(XL(S));"{$}"
4520 IF S>0 AND S<4 THEN FOR I=1 TO 4-S:PRINT TAB(XL(S))"{'}":NEXT
4545 IF S>0 THEN PRINT TAB(XL(S))LEFT$("{'$$$$}",6-S)
4565 IF S=4 THEN PRINT TAB(XL(S))"{' ' 2°left down '}"
4570 Q$="":IF S>0 THEN Q$=MID$("{3°right '}",S)
4572 IF S<4 THEN FOR I=0 TO N2(S)+1:PRINT TAB(XL(S))"{'}"Q$:NEXT
4575 IF S>0 THEN PRINT TAB(XL(S))LEFT$("{'####}",6-S)
4595 IF S>0 AND S<4 THEN FOR I=1 TO 4-S:PRINT TAB(XL(S))"{'}":NEXT
4596 IF S=0 THEN PRINT TAB(XL(S))"{#}"
4597 PRINT "{home}"
4599 RETURN
4700 DN=YL(S):GOSUB 7000
4715 IF S=0 THEN PRINT TAB(XL(S));"{M}":GOTO 4730
4720 FOR I=1 TO 5-S:PRINT TAB(XL(S)+I)"{M}":NEXT
4730 DN=N2(S)+2:GOSUB 7100
4735 IF S=0 THEN PRINT TAB(XL(S))"{N}":GOTO 4749
4740 FOR I=0 TO 4-S:PRINT TAB(XL(S)+5-S-I)"{N}":NEXT
4749 PRINT "{home}"
4750 RETURN
4800 GOSUB 4200
4815 DN=YL(S)+5-S:GOSUB 7000
4820 PRINT TAB(20);
4825 IF S<4 THEN FOR I=1 TO XR(S)-21:PRINT "{#}";:NEXT
4830 PRINT "{P left down}";
4835 IF S<4 THEN FOR I=1 TO N2(S):PRINT "{ ' down left}";:NEXT
4840 PRINT "{' : 2°left}";
4850 IF S<4 THEN FOR I=1 TO XR(S)-20-1:PRINT "{$ 2°left}";:NEXT
4851 PRINT "{home}"
4855 RETURN
4900 IF S=0 THEN J=5:GOTO 4910
4902 J=0
4910 GOSUB 4700
4915 DN=YL(S)+5-S:GOSUB 7000
4920 PRINT TAB(19);
4925 IF S<4 THEN FOR I=1 TO 18-XL(S)-5+S+J:PRINT "{# 2°left}";:NEXT
4930 PRINT "{O left down}";
4935 IF S<4 THEN FOR I=1 TO N2(S):PRINT "{% down left}";:NEXT
4940 PRINT "{L}";
4950 IF S<4 THEN FOR I=1 TO 18-XL(S)-5+S+J:PRINT "{$}";:NEXT
4951 PRINT "{home}"
4955 RETURN

```

```

5000 IF S=0 OR S=4 THEN RETURN
5010 DN=18:GOSUB 7000
5020 IF S=3 THEN PRINT TAB(20);"{3°up}."
5030 IF S=2 THEN PRINT TAB(19);"{rvs up ) off ) 2°left down &N up
      %}"
5040 IF S=1 THEN PRINT TAB(18);"{down}_{$$$ 4°left down N}__{N%
      5°left down &&&}__{% 5°left down &&N}"
5050 RETURN
5100 IF S=0 OR S=4 THEN RETURN
5110 DN=20:GOSUB 7000
5120 IF S=3 THEN PRINT TAB(19);"{5°up rvs "" up 2°left off ""}"
5130 IF S=2 THEN PRINT TAB(16);"{3°up rvs )}_.....{<}"
5141 IF S=1 THEN PRINT TAB(13);"{rvs )}_.....{<}"
5142 IF S=1 THEN PRINT TAB(12);"{rvs )}_.....{<}"
5150 RETURN
6000 K=TI:FOR I=1 TO 25:POKE 245,22:PRINT :
      PRINT TAB(15)"DELICIOUS!":FOR J=1 TO 50:NEXT J
6010 POKE 245,22:PRINT :PRINT TAB(15)"_.....":FOR J=1 TO 50:
      NEXT J,I
6100 GOSUB 2000
6110 PRINT TAB(10);"{up}ANOTHER?_";
6115 SW=1:TM=0
6120 GET T$:IF T$<>" " THEN 6200
6130 IF TI>TM THEN PRINT MID$("?_",SW,1);"{left}";:TM=TI+15:SW=3-SW
6140 GOTO 6120
6200 IF T$="Y" OR T$=CR$ THEN PRINT "YES":GOTO 200
6210 IF T$="N" THEN PRINT "NO{down}":END
6220 GOTO 6120
7000 PRINT "{home}";
7100 IF DN>0 THEN FOR Z=1 TO DN:PRINT "{down}";:NEXT
7110 RETURN
62000 PRINT "{clr down}";TAB(15-LEN(PG$));PG$;"_BY_C._T._NADOVICH"
62010 PRINT "{down}_.....CURSOR_#13_..COPYRIGHT_(C)_1981"
62020 FOR I=1 TO 13:PRINT "{@@@}";:NEXT:PRINT
62030 PRINT "{down}FIND_YOUR_WAY_THROUGH_A_PERSPECTIVE_MAZE"
62032 PRINT "{down}YOUR_CONTROLS_ARE:"
62034 PRINT "{down}_..8_..MOVE_FORWARD"
62035 PRINT "{down}_..4_..TURN_LEFT"
62036 PRINT "{down}_..6_..TURN_RIGHT"
62038 PRINT "{down}_..2_..TURN_AROUND"
62040 PRINT "{down}_..?_..HELP!"
62049 PRINT "{3°down}PRESS_{rvs}RETURN{off}_..TO_BEGIN"
62050 GET T$:IF T$="" THEN 62050
62060 CLR:QK=525:CR$=CHR$(13):IF PEEK(50000) THEN QK=158
62070 PRINT "{clr}":GOTO 100

```

GAMES OF RISK



MAZE

By Howard Arrington

So you want to be rich, do you? Would you be satisfied with numerous 24-carat gold bars? If so, this game is for you.

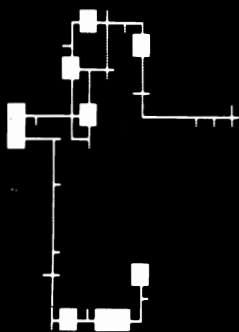
In MAZE, the PET constructs a maze of passages and rooms, and at your command, fills the rooms with gold. It doesn't do this secretly, either. You're free to watch if you like. When all the gold is stowed away, you get to venture in after it.

There is one minor hitch. When all the gold is safely in place, the map you were studying vanishes. You will move about by using the numeric keypad (8 moves up, 4 left, 6 right, and 2 down). As you travel, those parts of the maze

that you visit will become visible again. When you enter a room with gold in it, the value of the gold will be announced, and the gold added to your hoard. Beware, if a room has several caches of gold in it, you'll only find one at a time. You must leave the room and re-enter it to find another.

You may wander as long as you like, even after you've found all the gold. To stop your travels, press [RETURN]. If you despair of finding the gold, press [?], and the PET will show you where it all is (but won't let you in after it). One more thing — you're being timed, so don't waste time admiring your newly found riches.

```
■ TREASURE # 4
HIDDEN 122239 GOLD PIECES
WEALTH 17819 GOLD PIECES
```



```
0 CLR:POKE 59468,12
6 REM MAZE COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81 RJ
90 PG$="MAZE":NM$="8":GOSUB 62000
100 GOSUB 900:Q=Q-800:GOSUB 810
110 CLR:GOSUB 60400:GOSUB 900:Q1=WD*20:Q2=CRT+WD*5:
GP$="GOLD_PIECES"
200 SS=50:DIM A$(SS),C(12),D$(12,4),R$(8,4),B$(4,13),W$(4)
220 FOR I=0 TO 12:READ C(I):NEXT
230 DATA 32,64,107,109,110,125,112,113,114,115,91,93,160
250 FOR I=1 TO 4:FOR J=1 TO 13:READ B$(I,J):NEXT:NEXT
```



```

260 DATA 12,12,12,11,11,11,11,11,3,5,2,7,9,10,12,12,12,1,1,1,1,3,6,2,
    7,8,10
280 DATA 12,12,12,1,1,1,1,4,5,7,8,9,10,12,12,12,11,11,11,11,4,6,2,
    8,9,10
400 FOR I=1 TO 12:FOR J=1 TO 4:READ D%(I,J):NEXT:NEXT
410 DATA 0,1,1,0,1,0,1,1,0,0,1,1,1,1,0,0,0,1,0,1,1,0,1,0,0,1,1,1,1,
    1,1,0
420 DATA 1,1,0,1,1,1,1,1,1,1,0,0,1,1,1,1,1
430 FOR I=1 TO 4:READ D(I),W$(I):NEXT
435 DATA 40,SOUTH,-1,WEST,1,EAST,-40,NORTH
436 D(1)=WD:D(4)=-WD
440 BL$="....."
500 S=1+INT(RND(1)*Q1):E=10:P=91:J%=1
510 PRINT "{clr}NOW CONSTRUCTING MAZE.":Z=WD*.4:K=0
516 FOR J=Q TO Q+Q1-1:POKE J,0:NEXT
520 L=S:S1=L:A%(1)=L:POKE L+Q,E:POKE L+Q2,P
610 M=INT(RND(1)*4)+1:F=1
620 L=S+D(M):IF L<0 THEN L=L+Q1
626 IF L>=Q1 THEN L=L-Q1
630 P=PEEK(L+Q2):IF P>32 OR (P=32 AND D%(E,M)=0) THEN
    M=(M AND 3)+1:F=F+1:IF F<5 THEN 620
640 IF F=5 THEN 1200
700 F=INT(RND(1)*13)+1:IF F>9 THEN GOSUB 3000
720 E=B%(M,F):POKE L+Q2,C(E):S=L:POKE L+Q,E:IF E=12 THEN Z=Z-1
733 IF Z>0 THEN F=1:GOTO 620
735 IF K=0 THEN PRINT "{home}PRESS ANY KEY TO STOP CONSTRUCTION.":
    K=1
737 GET M$:IF M$="" THEN F=1:GOTO 620
740 GOTO 5000
800 Q=PEEK(QM)+256*PEEK(QM+1):RETURN
810 H=INT(Q/256):L=Q-256*H:POKE QM+1,H:POKE QM,L:RETURN
900 QM=134:QK=525:IF PEEK(50000) THEN QM=52:QK=158
910 GOTO 800
1200 S=A%(J%):IF S=0 THEN 5000
1230 P=PEEK(S+Q2):E=PEEK(S+Q):A%(J%)=0:J%=J%-1:GOTO 610
3000 IF J%<SS THEN J%=J%+1:A%(J%)=L
3030 RETURN
5000 PRINT "{home}";BL$:PRINT "{up}NOW HIDING THE GOLD..."
5005 TR=0:W=0:W1=0:FOR I=1 TO 25
5010 L=INT(RND(1)*Q1):IF PEEK(L+Q)<>12 THEN 5010
5020 A%(I)=L:POKE L+Q2,164:L=INT(RND(1)*10000)+1:TR=TR+L:A%(I+25)=L
5030 PRINT "{home 2°down}HIDDEN";TR;TAB(14);GP$:NEXT I
5100 PRINT "{home}PRESS {rvs}RETURN{off} TO BEGIN:"
6000 GET M$:IF M$="" THEN 6000
6018 L=S1:PRINT "{clr}":TI$="000000"
6030 E=PEEK(L+Q):P=C(E):P1=P
6040 POKE 32768,P:GET A$
6042 POKE L+Q2,P1
6045 IF A$="?" THEN 7000
6047 IF A$="Q" OR A$=CHR$(13) THEN 7100
6050 IF A$="" THEN P1=(P1+128) AND 255:GOTO 6040

```

```

6060 POKE L+Q2,P
6080 J=ASC(A$)-48:IF J<2 OR J>8 OR (J AND 1) THEN 6040
6100 J%=J/2:J=L+D%(E,J%)*D(J%)
6102 IF J<0 THEN J=J+Q1
6104 IF J>=Q1 THEN J=J-Q1
6110 F=PEEK(J+Q):PRINT "{home 2°right}";W$(J%);"....."
6120 IF D%(F,5-J%)=0 THEN PRINT "{home}";TAB(10);"BLOCKED":GOTO 6040
6130 L=J:IF F=12 THEN GOSUB 8000
6140 GOTO 6030
7000 PRINT "{home}..**.HERE'S.THE.MAZE:..**"
7010 T$=TI$:FOR J=0 TO Q1-1
7015 E=PEEK(J+Q):F=PEEK(J+Q2)
7016 I=128:IF F<>32 THEN I=-58*(E=12)
7017 IF E=12 THEN I=-I
7020 POKE J+Q2,C(E)+I:NEXT J
7030 FOR I=1 TO 25:IF A$(I)=-1 THEN 7070
7040 T=Q2+A$(I):F=PEEK(T):IF F=32 OR F=36 THEN POKE T,36
7050 IF F<>32 AND F<>36 THEN POKE T,164
7070 NEXT I
7080 TI$=T$
7100 T$=TI$:PRINT "{home down}...TIME.";LEFT$(T$,2);": ";MID$(T$,3,
2);": ";RIGHT$(T$,2);
7110 PRINT "....."
7200 PRINT "{home}";BL$:PRINT "{up}WANT.TO.PLAY.AGAIN?.";
7210 GOSUB 50000
7220 IF YN THEN FOR I=1 TO 500:NEXT I
7240 Q=Q+Q1:GOSUB 810:CLR
7250 PRINT "{4°down}"
7260 END
8000 FOR I=1 TO 25:IF A$(I)=L THEN 8100
8020 NEXT I:RETURN
8100 A$(I)=-1:I=A$(I+25):TR=TR-I
8105 W1=W1+1:PRINT "{home 2°right}TREASURE.#";W1
8110 PRINT "FOUND.";I;TAB(14);GP$
8120 IF TR>0 THEN PRINT "HIDDEN";TR;"{left}..{left}";TAB(14);GP$
8125 IF TR<=0 THEN PRINT "YOU.FOUND.ALL.THE.HIDDEN.GOLD!"
8140 W=W+I:PRINT "WEALTH";W;TAB(14);GP$
8150 FOR I=1 TO 500:IF PEEK(QK)<>0 THEN 8160
8155 NEXT I
8160 PRINT "{home down}";BL$
8170 RETURN
50000 YN=1
50010 SW=1:TM=0
50020 GET T$:IF T$<>" " THEN 50100
50030 IF TI>TM THEN PRINT MID$("?.",SW,1);"{left}";:SW=3-SW:TM=TI+30
50040 GOTO 50020
50100 IF T$="Y" THEN YN=1:GOTO 50150
50110 IF T$="N" THEN YN=0:GOTO 50150
50120 IF T$<>CHR$(13) THEN 50030
50150 PRINT MID$("NO.YES",1+3*YN,3):RETURN
60200 IF PN=0 THEN 60250

```

```

60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{TO}_";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTINUE:"
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN PRINT "{clr}":GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_CURSOR_#";NM$;
    "_.(C).1981....PAGE";PN
60270 GOSUB 60500:RETURN
60400 CRT=32768:WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60415 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;
    "_BY_HOWARD_ARRINGTON"
62002 PRINT "{down}.....CURSOR_#8..COPYRIGHT_(C).1981"
62004 GOSUB 60500
62010 PRINT "{down}FIND_THE_HIDDEN_TREASURES"
62020 PRINT "{2°down}WANT_DIRECTIONS?_";:YN=0:GOSUB 50010
62030 IF YN=0 THEN PRINT "{clr}";:CLR:POKE 59468,12:GOTO 100
62300 LP=3:GOSUB 60200
62310 PRINT "THE_OBJECT_OF_THIS_GAME_IS_TO_FIND_ALL"
62320 PRINT "THE_GOLD_HIDDEN_IN_A_MAZE."
62360 PRINT "{down}FIRST,_THE_SCREEN_IS_CLEARED,_AND_CON-"
62370 PRINT "STRUCTION_OF_THE_MAZE_IS_BEGUN._AT_ALL"
62375 PRINT "TIMES,_YOU_CAN_SEE_EXACTLY_HOW_DEVIOUS"
62376 PRINT "THE_MAZE_HAD_BECOME..WHEN_THE_MAZE_IS{ }"
62380 PRINT "AS_COMPLEX{ } (OR_AS_SIMPLE)_AS_YOU_LIKE,"
62390 PRINT "PRESS{ _ rvs}RETURN{off}_{TO}_STOP_CONSTRUCTION."
62420 PRINT "{down}WHEN_THE_MAZE_IS_COMPLETE,_THE_PET"
62430 PRINT "HIDES_A_NUMBER_OF_GOLD_PIECES_{(rvs)}${off})_IN"
62440 PRINT "VARIOUS_PLACES_IN_THE_MAZE."
62475 GOSUB 60200
62480 PRINT "ONCE_ALL_THE_GOLD_IS_HIDDEN,_YOU_MAY"
62490 PRINT "BEGIN_YOUR_SEARCH_BY_PRESSING_{rvs}RETURN{off}."
62500 PRINT "THE_MAZE_WILL_VANISH,_AND_YOU_WILL_FIND"
62510 PRINT "YOURSELF_AT_THE_START_OF_THE_MAZE."
62520 PRINT "YOU_CAN_MOVE_BY_PRESSING_THE{ _
    rvs}8{off},_{rvs}4{off},_{ }8"
62530 PRINT "{rvs}6{off},_AND_{rvs}2{off}_KEYS._{rvs}8{off}
    _MOVES_UP,_{rvs}4{off}_LEFT,{ }"
62540 PRINT
    "{rvs}6{off}_RIGHT,_AND_{rvs}2{off}_MOVES_DOWN...WHEN-_4{ @[@}6"
62550 PRINT "EVER_YOU_STUMBLE_OVER_SOME_GOLD,{ }"
62560 PRINT "ITS_VALUE_WILL_BE_ADDED_TO_YOUR.....2"
62570 PRINT "WEALTH..BEWARE,_HOWEVER!_SEVERAL"
62580 PRINT "PILES_OF_GOLD_PIECES_MAY_BE_HIDDEN_AT"
62590 PRINT "ANY_GIVEN_SPOT,_AND_YOU_MUST_RE-ENTER"
62600 PRINT "THE_HIDING_PLACE_TO_FIND_THE_OTHERS."
62610 GOSUB 60200
62620 PRINT "IF_YOU_FIND_ALL_THE_GOLD,{ }YOU_WILL_BE"

```

```
62630 PRINT "TOLD_SO..YOU_WILL_BE_FREE_TO_CONTINUE"
62640 PRINT "EXPLORING_THE_MAZE, HOWEVER..AT_ANY"
62650 PRINT "TIME, YOU_CAN_STOP_EXPLORING_BY_PRESS-"
62660 PRINT "ING{ rvs}RETURN{off}..IF_YOU_DESPAIR_OF_EVER"
62670 PRINT "FINDING_ALL_THE_GOLD, PRESS_{rvs}?{off}, AND_YOU"
62680 PRINT "WILL_BE_SHOWN_THE_MAZE{ } (AND_THE_GOLD)"
62690 PRINT "AGAIN."
62750 PRINT TAB(11) "_{3°down rvs}HAPPY_HUNTING!{off 2°down}"
62760 GOSUB 60200
```



DUNGEON

By Brian Sawyer

The evil magician Trent has transported you into the depths of some long-abandoned dungeons. The dungeons have been taken over by a large collection of monsters, each with its own hoard of gold. Trent has agreed to release you from his power if (and only if) you somehow get *all* the gold away from the monsters.

The dungeons themselves are the usual run-of-the-mill dungeons — dank, dark, dismal, and decrepit. These dungeons have a novelty, however. Being long-abandoned, several passages have caved in, blocking the route to the rooms they once led into. Trent has foreseen this, and has given you a way to pass through walls. Also, the various creatures have been there a long time (feeding on those unwise enough to come in seeking gold) and no longer fit through the doorways. Finally, the spell Trent used to put you there won't permit you to leave. Only *he* can get you out.

On the screen, you'll see a map of the dungeon, drawn as you travel about (your movement is controlled by the numeric keypad). Your position is marked with a dot. Gold, as you find it, is marked with a "G." The various sorts of creatures are marked with various sorts of markers. Open space is shown as white, and doors as grey squares. Walls are shown as black.

Again, nothing will show up on the screen until you find it in the dungeon.

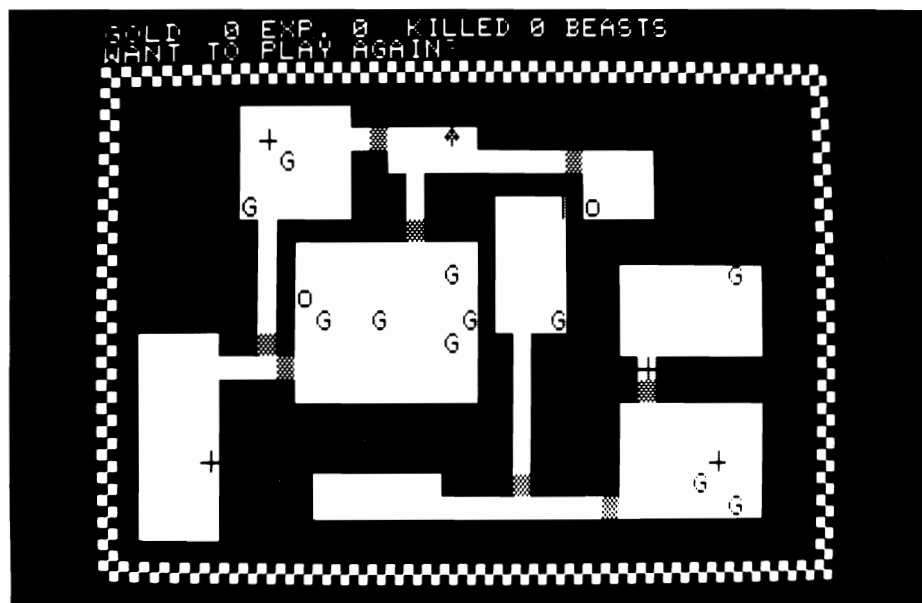
In addition to the map, lines of information are kept on the screen. They show your hit points, your experience points, and how much gold you have.

Hit points measure how much strength you have (zero means you're dead). Wandering about doing nothing uses up a few hit points; battling a monster uses up a great many. You may rest if your hit points get too low, but do so in a safe place. The monsters find resting humans to be just as tasty as any other kind.

Experience points indicate how good you are at surviving. The more creatures you defeat, the higher your experience points go. The more experience points you get, the easier it is to defeat creatures of a given strength.

To move, use the numeric pad in the usual fashion. To pick up gold, move on top of it. To attack a creature, move on top of it. If you want to move through the wall, press [SHIFT] with the appropriate number. (Walking through walls uses up more energy, and more hit points, than walking through open space.) If you want to quit (and suffer whatever *other* torments Trent can devise), press [Q].

As the Boy Scouts say, be prepared.



```

0 CLR:POKE 59468,12
6 REM DUNGEON COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81
90 PG$="DUNGEON":NM$="15":GOSUB 62000
100 PRINT CHR$(142):RS=23:CS=WD:SZ=RS*CS:BL=(25-RS)*CS:RS=RS-1:
    CS=CS-1
110 DIM DG((SZ+BL)/5),DR(9):REM DG() USED TO GRAB MEMORY
115 P=WD:FOR I=2 TO 8 STEP 3:FOR N=-1 TO 1:DR(I+N)=P+N:NEXT N:
    P=P-WD:NEXT I
120 ER$="{home down}.....{home
    down}"
130 E2$="{home}.....{home}"
140 A=0:C0=0:CC=0:E1=0:GN=0:I=0:IU=0:L=0:N=0:N1=0:O1=0:P=0:PW=0:
    QQ=0:R0=0
150 Q=0:SX=0:TS=0:U=0:V=0:V1=0:Y=0:ZC=0:ZT=0
160 EX$="EXP.":HP$="HIT_PTS.":IF WD>40 THEN EX$="EXPERIENCE":
    HP$="HIT_POINTS"
170 L$="":E$=""
180 PRINT "{clr}SETTING UP..."
190 HP=50:MG=0:EX=0:PX=0:HG=0:Z=0:FG=0:K1=0:E=0:S=0:W=160:ET=160
200 TI$="000000":TM=TI+3600*WD/40
210 AX=CRT:TS=PEEK(QA)+256*PEEK(QA+1)+7+BL
220 FOR I=TS+WD TO TS+SZ-WD:POKE I,32:NEXT I
230 GOSUB 380
240 TS=TS-BL
250 L=INT(RND(1)*SZ+TS):IF PEEK(L)<>160 THEN 250
260 TM=0:GOSUB 1410:L=L+AX-TS:W=PEEK(L):GOSUB 600:POKE L,209:W=160
270 IF PEEK(TS+BL)<>127 THEN PRINT "{home rvs}ABORT!!":GOTO 1360
275 POKE QK,0:POKE QP,255:GOSUB 1240
280 HP=HP-.15-2*SX:IF HP<0 THEN 1190
300 IF (PEEK(L+A)=32) AND (SX<>1) THEN 270
310 IF PEEK(L+A)=127 THEN 270
320 POKE L,W:L=L+A:W=PEEK(L):POKE L,209:GOSUB 600:POKE L,209
330 IF W=135 THEN GOSUB 1200
340 IF W>=214 AND W<=219 THEN GOSUB 1000
350 IF E>0 THEN S=S+1
360 IF S>1 THEN GOSUB 830
370 GOTO 270
380 PRINT INT((TM-TI)/60);"{left}..{up}"
400 W=INT(RND(1)*9+2):L=INT(RND(1)*9+2)
410 R0=INT(RND(1)*(RS-L-1))+1:C0=INT(RND(1)*(CS-W-1))+1:
    P=TS+WD*R0+C0
420 IF P+WD*L+W>=TS+SZ THEN 530
430 FOR N=0 TO L+1:FOR N1=0 TO W+1:IF PEEK(P+(N*WD)+N1)<>32 THEN
    530
440 NEXT N1,N
450 FOR N=1 TO L:FOR N1=1 TO W:POKE P+(N*WD)+N1,160:NEXT N1,N
460 FOR N=P+2+(L+1)*WD TO TS+SZ-1 STEP WD
465 Q=PEEK(N):POKE N,0
470 IF Q=160 THEN 485
475 FOR A=1 TO 100:NEXT A

```

```

480 POKE N,Q:NEXT N:GOTO 490
485 FOR N1=P+2*WD TO N STEP WD:POKE N1,160:NEXT:POKE N1-2*WD,102
490 FOR N=P+2*WD+1+W TO P+3*WD+1+W:IF (N-TS)/WD=INT((N-TS)/WD) THEN
    520
500 IF PEEK(N)=160 THEN FOR N1=P+2*WD+1 TO N-1:POKE N1,160:NEXT:
    POKE N1-1,102:GOTO 520
510 NEXT
520 S=INT(RND(1)*L)+1:A=INT(RND(1)*W+1):POKE P+A+S*WD,
    INT(RND(1)*6+214)
530 IF TI<TM THEN 380
540 FOR N=1 TO 11
550 U=INT(RND(1)*SZ)+TS:IF PEEK(U)<>160 THEN 550
560 POKE U,135:HG=HG+1:NEXT
570 FOR R0=0 TO RS:POKE TS+WD*R0,127:POKE TS+WD*R0+CS,127:NEXT R0
580 FOR C0=0 TO CS:POKE TS+C0,127:POKE TS+C0+WD*RS,127:NEXT C0
590 RETURN
600 GN=0
640 FOR N=-WD TO WD STEP WD:FOR N1=-1 TO 1:IF N=0 AND N1=0 THEN 820
650 Y=L-AX+N+N1:V=PEEK(Y+TS):POKE Y+AX,V
660 IF V<135 OR V=160 THEN 820
670 V=V-128:IF V<>7 THEN 710
680 K1=1+K1+INT((MG+1)*(RND(1)))
690 GN=GN+1:IF GN>FG THEN GOSUB 1410:PRINT "GOLD_IS_NEAR!":
    GOSUB 1430:FG=HG+1
700 GOTO 820
710 V1=V+128:S=0:POKE Y+TS,160
720 IF V=86 THEN E$="SPIDER":I=3
730 IF V=87 THEN E$="GRUE":I=7
740 IF V=88 THEN E$="DRAGON":I=1
750 IF V=89 THEN E$="SNAKE":I=2
760 IF V=90 THEN E$="NUIBUS":I=9
770 IF V=91 THEN E$="WYVERN":I=5
780 I=INT(RND(1)*HP+(PX/I)+HP/4)
790 IF E>0 THEN POKE TS+E,QQ
800 QQ=V+128:E=Y
810 GOSUB 1410:PRINT "A_"E$;"_WITH";I;"POINTS_IS_NEAR.":GOSUB 1430:
    CC=I
820 NEXT N1:NEXT N:FG=GN:RETURN
830 O1=0:A=0:E1=E+AX:IF ABS(E1+WD-L)<ABS(E1-L) THEN A=WD
840 IF ABS(E1-WD-L)<ABS(E1-L) THEN A=-WD
850 IF (PEEK(E+A+TS)=209) OR (PEEK(E+A+TS)>128) THEN O1=O1+A
860 IF ABS(E1-1-L)<ABS(E1-L) THEN A=-1
870 IF ABS(E1+1-L)<ABS(E1-L) THEN A=1
880 IF (PEEK(E+O1+A+TS)=209) OR (PEEK(E+O1+A+TS)>128) THEN O1=O1+A
890 A=O1:IF E1+A=L THEN 960
900 IF E1+A<L THEN 940
910 IF A=DR(3) THEN A=DR(1)
920 IF A=DR(9) THEN A=DR(7)
930 GOTO 960
940 IF A=DR(1) THEN A=DR(3)
950 IF A=DR(7) THEN A=DR(9)

```

```

960 IF PEEK(E+A+TS)<128 THEN POKE E1,V1:RETURN
970 POKE E1,ET:E=E+A:ET=PEEK(E+AX):POKE E+AX,V1:IF E+AX=L THEN
    ET=160:GOSUB 1000
980 RETURN
990 QQ=V+128:E=Y
1000 GOSUB 1410:PRINT "{rvs}AN_ATTACK!!!":GOSUB 1430:PW=HP+EX
1010 HP=INT(HP-(RND(1)*CC/2+CC/4)):CC=INT(CC-(RND(1)*PW/2+PW/4))
1020 IF HP<1 THEN 1190
1030 IF (CC/(HP+1)<2) THEN 1090
1040 GOSUB 1410:PRINT "THE_\"E$\"_WILL_LEAVE..":GOSUB 1430
1050 GOSUB 1410:PRINT "IF_YOU_GIVE_IT_HALF_YOUR_GOLD.":GOSUB 1430
1060 GOSUB 1410:PRINT "WILL_YOU_CONSENT_TO_THIS_(Y_OR_N)_";
1070 GOSUB 1500
1080 IF L$="Y" THEN MG=MG-MG/2:W=160:E=0:S=0:POKE L,209:RETURN
1090 IF CC>1 THEN 1160
1100 GOSUB 1410:W=160:S=0:E=0:POKE L,209:PRINT "THE_\"E$\"_IS_DEAD!":
    GOSUB 1430
1110 EX=EX+I:Z=Z+1
1120 IF EX<PX*2 THEN 1180
1130 PX=EX:HP=HP*3
1140 GOSUB 1410:PRINT "YOUR_HIT_PTS._HAVE_BEEN_RAISED.":GOSUB 1430
1150 GOTO 1180
1160 GOSUB 1410:PRINT "THE_\"E$\"_HAS\"CC\"HIT_POINTS":GOSUB 1430
1170 GOSUB 1340
1180 RETURN
1190 GOSUB 1410:PRINT "YOU'RE_DEAD!":GOSUB 1430:GOTO 1350
1200 MG=MG+K1:GOSUB 1410:PRINT "YOU_FOUND\"K1\"GOLD_PIECE";:
    IF K1<>1 THEN PRINT "S";
1205 PRINT "!":GOSUB 1430
1210 POKE L-AX+TS,160:W=160
1220 HG=HG-1:IF HG>0 THEN RETURN
1230 GOTO 1350
1240 IF IU=0 THEN GOSUB 1430
1250 GOSUB 1340
1260 IF IU THEN IF TI>TM THEN GOSUB 1410:PRINT "YOU_MAY_MOVE."
1270 GET L$:IF L$="" THEN 1260
1280 A=ASC(L$) AND 127: SX=PEEK(QS)
1290 IF A=ASC("5") THEN HP=HP+1+SQR(EX/HP)
1300 IF A>48 AND A<58 THEN A=DR(A-48):TM=0:GOSUB 1410:RETURN
1320 IF L$="Q" THEN 1350
1330 GOTO 1250
1340 PRINT E2$;HP$;INT(HP+.5);EX$;EX;"_GOLD";MG;"_...":RETURN
1350 GOSUB 1410:PRINT E2$;"GOLD:";MG;EX$;EX;"_KILLED";Z;"BEASTS"
1360 FOR N=BL TO SZ-1+BL:A=PEEK(TS+N):POKE AX+N,A:NEXT
1375 GET L$:IF L$<>"" THEN 1375
1380 GOSUB 1410:PRINT "WANT_TO_PLAY_AGAIN";
1390 GOSUB 1500:IF L$<>"N" THEN 180
1400 TM=0:GOSUB 1410:PRINT "{up}";:END
1410 IF IU THEN IF TI<TM THEN 1410
1420 PRINT ER$;:IU=0:RETURN
1430 TM=TI+3*60:IU=1:RETURN

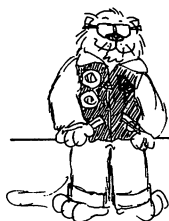
```



```

1500 ZT=0:ZC=1
1510 GET L$:IF L$<>" THEN 1550
1520 IF TI>ZT THEN PRINT MID$("?",ZC,1);"{left}";:ZT=TI+30:ZC=3-ZC
1530 GOTO 1510
1550 PRINT "?";L$:RETURN
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 CRT=32768:QA=126:QK=525:QP=515:QS=516
60410 IF PEEK(50000) THEN QA=44:QK=158:QM=52:QS=152
60420 WD=40:POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:
PRINT CHR$(142)
60430 RETURN
60500 FOR I=1 TO 13:PRINT "@{}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;".BY.BRIAN.SAWYER"
62010 PRINT "{down}.....CURSOR.#15..COPYRIGHT.(C)..1981"
62020 GOSUB 60500
62030 PRINT "{down}SEARCH.FOR.GOLD.IN.THE.ANCIENT.RUINS"
62080 PRINT "{2°down}PRESS.{rvs}RETURN{off}.TO.BEGIN"
62090 GET T$:IF T$="" THEN 62090
62100 GOTO 60300

```



EVEREST

By Brian Sawyer

The object of EVEREST is to climb to the top of Mt. Everest before Mother Nature kills you. The variety of ways your fate could be sealed includes total exhaustion, freezing to death, and being hit by an avalanche. Why play a game where the odds seem to be stacked against you so unfairly? Because it's there.

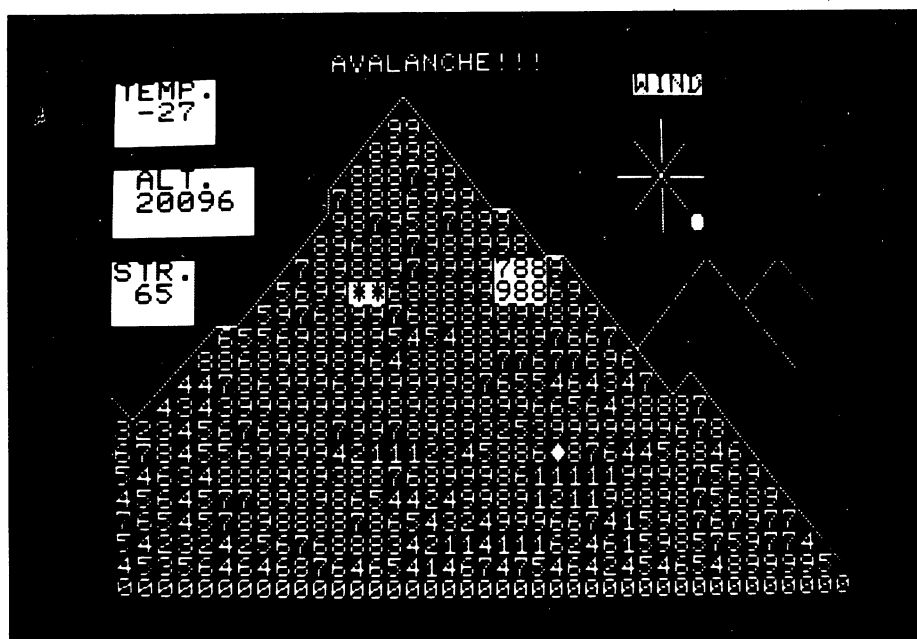
The mountain is covered with numbers between zero and nine that indicate the thickness of the snow in a particular spot. A nine means that the snow is very deep, and the amount of energy you expend climbing is high, as opposed to a one, which requires much less effort. As you climb up the mountain, traveling from snow patch to snow patch (using the numeric pad), you must be aware of the avalanches and blizzards that may come your way.

A blizzard is represented by a patch of white, and it moves in the direction that the wind blows. Since there is a weather vane in the top right

corner of the screen, you will be able to gauge the path of the blizzard, and perhaps avoid being caught in it. Since a blizzard deposits fresh snow on the mountain, it causes the climbing to be more difficult, and so the snow-depth numbers increase when a blizzard passes over them.

An avalanche is represented by "***" and is also best avoided. However, when an avalanche moves down a mountain, it shears away loose snow, and leaves behind an icy crust. This makes the going easier, and the numbers are consequently decreased when an avalanche passes over them.

Your climb ends when you either reach the top or succumb to the harsh elements. If you don't succeed in conquering the great mountain, the PET will consider the distance that you covered, rate your mountain climbing skill accordingly, and tempt you with another life for you to offer to the mountain.



```
0 CLR:POKE 59468,12
6 REM  EVEREST COPYRIGHT (C) 1981 THE CODE WORKS
10 REM  AS OF 10/24/81 GTF
90 PG$="EVEREST":NM$="15":GOSUB 62000
100 GN=1:O=1:R$="{off home}....."
```

```

110 PK=CRT+WD*23+29:SF=(12500/855)*(40/WD)
120 REM
150 PRINT "{clr down}"SPC(28){rvs}WIND{off left 2°down 2°left }
    down 2°left M\N down 4°left @@V@@ down 4°left N\M down 2°left
    }}"
160 TI$="000008"
170 M$=STR$(WD)+STR$(WD+1)+STR$(WD+2)+"_00_01_02"+STR$(-WD)+STR$(-WD+1)
175 M$=M$+STR$(-WD+2)
180 P=CRT+WD*23+20:P1=48:C=90
190 PRINT "{home 2°down}.....{NM}"
200 PRINT ".....{N}99{M}"
210 PRINT ".....{N}8998{M}"
220 PRINT ".....{N}998798{M}"
230 PRINT ".....{' }79986878{M$}"
240 PRINT ".....{N}9989587699{M}"
250 PRINT ".....{N}897987989798{M$}"
260 PRINT ".....{N}789999799996779{M}...../\_./\"
270 PRINT ".....{N}56998768889997769{M}...../..\_./\"
280 PRINT ".....{N}5976899768889898799{M}...../_.....\"
290 PRINT ".....{N}6556989895454889896567{MN}.....\"
300 PRINT ".....{N}886898989643889877676586{M}.....\"
310 PRINT ".....{N}4478699969999987655453247{MNM}"
320 PRINT "{MN}4343999999989898996656498887{M}"
330 PRINT "323456769987997898932539999999678{M}"
340 PRINT "6784556999942111234588698764456846{M}"
350 PRINT "54634888989388768999861111199987569{M}"
360 PRINT "456457789898654424998912119999875689{M}"
370 PRINT "7654578988887865432499966742698767977{M}"
380 PRINT "54232425676888342114111624626985759774{M}"
390 PRINT "453564646876465414674754642454654899995{M}"
400 PRINT "{up}2355645467899876543212345678999534887931";
405 IF WD=80 THEN PRINT "{down}"
410 PRINT "{up}0000000000000000000000000000000000000000";
420 PRINT "{home down rvs}TEMP.{down 5°left}.....{down
    5°left}....."
430 PRINT "{down rvs}_ALT_...{down 7°left}.....{down
    7°left}.....":POKE P,90
440 PRINT "{down rvs}STR.{down 4°left}.....{down 4°left $$$$}"
450 B=0:A=INT(RND(10)*WD*25)+CRT
460 IF PEEK(A)<48 OR PEEK(A)>58 OR PEEK(A+1)<48 OR PEEK(A+1)>58
    THEN 450
470 IF PEEK(A+WD)<48 OR PEEK(A+WD)>58 OR PEEK(A+WD+1)<48 OR
    PEEK(A+WD+1)>58 THEN 450
480 IF PEEK(A+WD+2)<48 OR PEEK(A+WD+2)>58 OR PEEK(A+2)<48 OR
    PEEK(A+2)>58 THEN 450
490 IF ABS(A-V)<4 OR (ABS(A-V)>37 AND ABS(A-V)<45) OR (ABS(A-V)>77
    AND ABS(A-V)<84) THEN 450
500 POKE A,PEEK(A)+128:POKE A+1,PEEK(A+1)+128:POKE A+WD,
    PEEK(A+WD)+128
510 POKE A+WD+1,PEEK(A+WD+1)+128:POKE A+2,PEEK(A+2)+128:
    POKE A+WD+2,PEEK(A+WD+2)+128

```

```

520 REM FORJJ=1TOO*1000:NEXT
530 GET G$:G=VAL(G$):IF G=0 THEN P2=0:GOTO 560
540 POKE P,P1:P2=VAL(MID$(M$,G*3-2,3))-1:IF PEEK(P+P2)>47 AND
    PEEK(P+P2)<58 THEN P=P+P2
550 P1=PEEK(P):POKE P,90:C=C-INT((P1-48)/3):REM DECREMENT STRENGTH
    ACCORD DNGR.
560 GOTO 1400
570 F=INT((P-CRT)/WD)-45+INT(RND(10)*5-2):PRINT "{home 2°down
    right rvs}"F
580 E=E+1:X=1-(E AND 1)
590 H=INT((PK-P)*SF)+16500:PRINT "{3°down rvs}"H:IF C<0 THEN C=0
591 REM H=INT(((33717-P)*12500)/855)+16500:PRINT"{3°down rvs}"H:
    IFC<0THENC=0
600 PRINT "{3°down rvs}"INT(C);:IF C<10 THEN PRINT "{rvs left}_"
610 IF P=A OR P=A+1 OR P=A+WD OR P=A+WD+1 OR P=A+WD+2 OR P=A+2 THEN
    GOSUB 1310
620 D=PEEK(A):D=D-128:GOSUB 1490:IF D<57 AND D>47 THEN D=D+X
630 POKE A,D
640 D=PEEK(A+1):D=D-128:GOSUB 1490:IF D<57 AND D>47 THEN D=D+X
650 POKE A+1,D
660 D=PEEK(A+WD):D=D-128:GOSUB 1490:IF D<57 AND D>47 THEN D=D+X
670 POKE A+WD,D
680 D=PEEK(A+WD+1):D=D-128:GOSUB 1490:IF D<57 AND D>47 THEN D=D+X
690 POKE A+WD+1,D
700 D=PEEK(A+WD+2):D=D-128:GOSUB 1490:IF D<57 AND D>47 THEN D=D+X
710 POKE A+WD+2,D
720 D=PEEK(A+2):D=D-128:GOSUB 1490:IF D<57 AND D>47 THEN D=D+X
730 POKE A+2,D
740 IF VAL(TI$)/11<>INT(VAL(TI$)/11) THEN 780
750 MN=INT(RND(10)*9+1)
760 W=VAL(MID$(M$,MN*3-2,3))-1
770 GOSUB 980
780 A=A+W:IF PEEK(A+1)>57 OR PEEK(A+1)<48 THEN 450
790 GOTO 490
800 FOR TY=1 TO 3:REM FREQ OF AVAL.
810 V=INT(RND(10)*WD*20+CRT)
820 IF PEEK(V)<>57 OR PEEK(V+1)<>57 THEN 840
830 GOTO 850
840 NEXT:GOTO 570
850 T=PEEK(V):T1=PEEK(V+1):POKE V,170:POKE V+1,170:IF P2=0 THEN 880
860 IF T=90 THEN T=P1:GOSUB 1380
870 IF T1=90 THEN T1=P1:GOSUB 1380
880 O=0:GOTO 570
890 PRINT "{home 11°right}AVALANCHE!!!"
900 PRINT "{home rvs 11°right}AVALANCHE!!!"
910 PRINT "{home 11°right}AVALANCHE!!!"
920 POKE V,T-S:POKE V+1,T1-S
930 S=1:V=V+WD:IF (PEEK(V)>57) OR (PEEK(V)<50) THEN S=0
940 IF V>CRT+WD*21+39 THEN PRINT R$:O=1:GOTO 570
950 IF (PEEK(V+1)>57) OR (PEEK(V+1)<50) THEN S=0
960 IF P=V-WD OR P=V-WD-1 THEN PRINT R$:GOSUB 1380

```

```

970 GOTO 850
980 POKE CRT+J,32
990 IF W=1 THEN J=WD*5+32
1000 IF W=-1 THEN J=WD*5+26
1010 IF W=WD THEN J=WD*8+29
1020 IF W=WD-1 THEN J=WD*7+27
1030 IF W=WD+1 THEN J=WD*7+31
1040 IF W=-WD-1 THEN J=WD*3+27
1050 IF W=-WD THEN J=WD*2+29
1060 IF W=-WD+1 THEN J=WD*3+31
1070 POKE CRT+J,81
1080 RETURN
1090 PRINT R$"{home rvs 8°right}OUT_OF_ENERGY!"
1100 FOR JJ=1 TO 2100:NEXT
1110 PRINT R$"{home 10°right}YOU'RE_DEAD!"
1120 GOTO 1160
1130 PRINT R$"{home rvs 6°right}YOU_HAVE_REACHED_THE_SUMMIT!"
1140 FOR JJ=1 TO 2100:NEXT
1150 PRINT "{home 5°right}YOU'VE_CONQUERED_MT._EVEREST!"
1160 TM=TI+350
1165 IF TI<TM THEN 1165
1170 C=INT(C/7)+INT((H-17000)/100):A$="ROTTEN":C=C-7:IF C>20 THEN
    A$="POOR"
1180 IF C>40 THEN A$="...FAIR..."
1190 IF C>60 THEN A$="GOOD":IF C>80 THEN A$="GREAT":IF C>100 THEN
    A$="EXCELLENT!"
1200 PRINT "{clr 5°down}SCORE:"ABS(C):
    PRINT "{down}YOUR_CLIMBING_WAS_"A$
1210 GET G$:IF G$<>" " THEN 1210
1220 IF GN>1 THEN
    PRINT "{down}AVERAGE_SCORE_FOR_"GN"GAMES:"INT((GC+C)/GN)
1230 PRINT "{2°down}WANT_TO_TRY_AGAIN?_";:GOSUB 60000:IF LEFT$(IN$,
    1)="N" THEN 1240
1235 GN=GN+1:GC=GC+C:GOTO 120
1240 END
1250 REM FALL
1260 POKE P,P1:P=P+WD:P1=PEEK(P):IF P>CRT+WD*24+39 THEN 1300
1270 IF P1=32 THEN 1100
1280 IF P1<50 THEN POKE P,90:RETURN
1290 POKE P,90:FOR N=1 TO 100:NEXT:GOTO 1260
1300 PRINT R$:GOTO 1100
1310 PRINT R$"{home 6°right rvs}YOU'RE_CAUGHT_IN_A_BLIZZARD!"
1320 FOR JJ=1 TO 2100:NEXT:PRINT R$
1330 C=C-15:IF C<1 THEN 1390
1340 POKE P,218:IF A1=0 THEN 1370
1350 IF T=90 THEN T=91
1360 POKE P,90:IF T1=90 THEN T1=91
1370 A1=0:RETURN
1380 PRINT R$"{home 6°right rvs}YOU_WERE_HIT_BY_THE_AVALANCHE!":
    A1=1:GOTO 1320
1390 PRINT R$"{home 9°right rvs}YOU_FROZE_TO_DEATH!":GOTO 1160

```

```

1400 IF P<32910 THEN 1130
1410 IF C<1 THEN PRINT "{home 10°down right rvs}0":GOTO 1090
1420 GET G$:IF G$<>" " THEN 1420
1430 IF P2=0 THEN C=C-.25:GOTO 1470:REM TIME PENALTY
1440 IF P>CRT+WD*23 THEN 1460
1450 C=C+INT(F/30):C=C-INT((H-15000)/3150):REM TEMP AND ALTITUDE
1460 IF INT(RND(10)*7)+(P1-48)=14 THEN GOSUB 1250
1470 IF O=0 THEN 890
1480 GOTO 800
1490 IF D<0 THEN D=D+128
1500 RETURN
60000 IN$="␣":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("␣{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "␣{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60240 GET T$:IF T$="" THEN 60240
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 CRT=32768:WD=40:PRINT "{clr}"
60410 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60430 RETURN
60500 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;"␣BY␣BRIAN␣SAWYER"
62010 PRINT "{down}␣␣␣␣␣␣CURSOR␣#15␣␣COPYRIGHT␣(C)␣1981"
62020 GOSUB 60500
62030 PRINT "{down}BECOME␣A␣MOUNTAINEER!"
62080 PRINT "{2°down}PRESS␣{rvs}RETURN{off}␣TO␣BEGIN␣␣";:GOSUB 60000
62090 GOTO 60300

```



GODZILLA



By Randall Lockwood

Godzilla is back for his annual rampage in Japan. You, as commander of all the armed forces, must stop him. As Godzilla refuses to listen to reason, you must kill him to halt his rampage. At your disposal are an army, a navy, an air force, ten missiles which can destroy several city blocks, and an atomic bomb.

On the screen of your control center, you'll see a map of Japan, divided into 25 sectors, in five rows and five columns. The rows and columns are numbered from 0 to 4. Sectors, in turn, are indicated by the numbers of the row and column they're in. For example, Sector 32 is at the intersection of row 3 and column 2. To control the mayhem, use the following commands:

- I** Information. You use this to find out the military strength of a sector, as well as its population.
- TA** Troop Attack. With this, you order troops to attack Godzilla. You may choose how many troops to use in the attack. Only troops in the same sector as Godzilla participate in the attack.
- SA** Ship Attack. *All* the ships in Godzilla's sector attack him.

- AR** Air Raid. You specify the number of planes which will attack Godzilla from the sector of your choice.
- M** Missile. This launches one of your missiles toward Godzilla.
- AB** Atomic Bomb. This will launch your atomic bomb at Godzilla. The bomb's explosion devastates the nine sectors surrounding Godzilla, so be cautious with this.
- TM** Troop Movement. With this, you may move your personnel from sector to sector. (You may only move land-based troops.)
- SM** Ship Movement. You may move one ship from one sector to another.
- Q** Quit. This ends the game, and cleans up after sound.

When giving movements, type the two sector numbers right next to each other, as 1201 for sector 12 to sector 01.

One final word: Godzilla doesn't die easily. You'll need all the resources at your command.



```

0 CLR:POKE 59468,12
6 REM  GODZILLA COPYRIGHT(C)1981 THE CODE WORKS
10 REM  10/24/81
100 Z=RND(-TI):GOTO 140
118 PRINT S$"{rvs}TOO_FAR"
120 FOR Z=1 TO 1500:NEXT:RETURN
130 PRINT C$S$"{rvs}CAN'T":GOSUB 120:RETURN
140 G$="GODZILLA":S=59464:POKE S,0:POKE S+3,16:POKE S+2,51:
    IF RR THEN 185
150 PRINT "{clr down}"TAB(7);G$;"_BY_R._LOCKWOOD"
152 PRINT "{down}.....CURSOR_#19_..COPYRIGHT_(C)_1981"
160 FOR I=1 TO 13:PRINT "{CCC}";:NEXT:PRINT :
    PRINT "{down}SAVE_TOKYO!"
170 PRINT "{2°down}PRESS_{rvs}RETURN{off}_TO_BEGIN"
180 GET A$:IF A$="" THEN 180
185 CRT=32768:WD=40:PRINT "{clr}"
186 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
190 DIM P(4,4),N(4,4),T(4,4),U$(4,4),S$(4,4),W$(4,4)
200 S$="{home 17°down 23°right}":K%=2500:K=1E7
205 C$=S$+".....{down 16°left}....."
210 B$=".....{9°left}":DF$="{down 3°right ? 3°left}"
220 FOR I=0 TO 2:N(I,3)=K:NEXT:N(3,0)=K:N(3,1)=K/1.5
230 N(3,2)=K:N(3,3)=K/.4:N(4,0)=K:N(4,1)=K
240 FOR A=0 TO 4:FOR B=0 TO 4:READ I:IF WD=80 THEN
    I=INT(I/40)*80+I-INT(I/40)*40
245 P(A,B)=I+CRT+WD:NEXT:NEXT
250 DATA 83,87,92,97,102,283,287,291,256,301,483,487,492,456,501,
    683,687,691
260 DATA 655,701,844,806,851,856,861
270 FOR A=0 TO 4:FOR B=0 TO 4:READ U$(A,B):NEXT:NEXT
280 DATA 0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,1,1,1,0,1,1,0,0,0
290 FOR A=0 TO 4:FOR B=0 TO 4:IF U$(A,B)=0 THEN 310
300 W$(A,B)=RND(1)*100:T(A,B)=INT(N(A,B)/80):GOTO 320
310 S$(A,B)=RND(1)*6+1:T(A,B)=S$(A,B)*350
320 NEXT:NEXT:M%=10:GOSUB 400
330 X=INT(RND(1)*5):Y=INT(RND(1)*5):GOSUB 570:GOTO 370
340 GD=GD-15:PRINT "{home 2°down}":PRINT SPC(23)G$"_AT:"X;Y:
    POKE PG,G$:PG=P(X,Y)
350 G%=PEEK(PG):POKE PG,135:FOR I=1 TO 7:POKE PG,7:POKE S,200:
    FOR Z=1 TO 70:NEXT
360 POKE PG,135:POKE S,80:FOR Z=1 TO 90:NEXT:NEXT:POKE S,0:RETURN
370 GOSUB 340:GOSUB 570:GOSUB 650:GOSUB 510:GOTO 370
400 PRINT "{clr}_";:FOR A=0 TO 4:PRINT A"_";:NEXT
410 PRINT :FOR I=0 TO 3:PRINT "{2°down}"CHR$(48+I)"{2°down}":NEXT:
    PRINT "{2°down}4{home 2°down}"
420 FOR I=1 TO 20:READ Z,Z$:PRINT SPC(Z)Z$:NEXT:RETURN
430 DATA 17,"{rvs}_{<}",17,"{rvs}..",17,"{rvs}.....{off 5}",16,
    "{rvs )}.....{off <}",15,"{rvs ) off 88<})",15,"{< rvs < down}"
440 DATA 16,"{rvs}_{<}",16,"{rvs}..",16,"{rvs}..",15,"{rvs )}..{off
    )}",14,"{rvs )}..",13,"{rvs )}....."
470 DATA 11,"{rvs}.....{off )}",11,"{rvs}.....T",6,"{rvs

```



```

    )}.....",4,"{rvs <}_{off )8 rvs}_{off 888)}"
490 DATA 7,"{rvs ) off}_{rvs}_{off })",3,"{rvs}_{off}_{rvs })",4,
    "{rvs}_{",4,"{)}"
510 IF TK>2E7 THEN 2020
520 IF GD>=K% THEN 1800
530 X1=X:X=X+2-(INT(RND(1)*3+1)):IF X>4 OR X<0 THEN X=X1:GOTO 530
540 Y1=Y:Y=Y+2-(INT(RND(1)*3+1)):IF Y>4 OR Y<0 THEN Y=Y1:GOTO 540
550 IF X=3 AND Y=3 THEN GOSUB 1940
560 GOSUB 1830:RETURN
570 PRINT "{home 4°down}":PRINT SPC(23)"INFO.FOR:"X;Y
580 IF N(X,Y)<1 THEN N(X,Y)=0
590 PRINT :PRINT SPC(23)"POP.:";B$ INT(N(X,Y))
600 IF T(X,Y)<0 THEN T(X,Y)=0
610 PRINT SPC(23)"TROOPS:"B$T(X,Y)
620 PRINT SPC(23)"SHIPS:"B$S%(X,Y)
630 PRINT SPC(23)"PLANES:"B$W%(X,Y):RETURN
638 PRINT C$$S$HOW.MANY"
640 PRINT S$DF$;:INPUT R$:R=VAL(R$):AB=R$="{?}":RETURN
650 IF N(3,3)<1000 THEN 1980
660 IF TK>2E7 THEN 2020
670 PRINT C$$S$COMMAND":GOSUB 640
680 IF AB THEN PRINT "{2°up}":GOTO 670
690 IF R$="I" THEN 1480
700 IF R$="TM" THEN 780
710 IF R$="SM" THEN 890
720 IF R$="TA" THEN 990
730 IF R$="M" THEN 1120
740 IF R$="AR" THEN 1240
750 IF R$="SA" THEN 1360
760 IF R$="AB" THEN 1500
765 IF R$="Q" THEN 2040
770 GOSUB 130:GOTO 650
780 GOSUB 638:TM=R:IF AB THEN 650
790 PRINT C$$S$FROM.TO"
800 GOSUB 640:IF AB THEN 650
810 GOSUB 1630:IF E<>0 THEN 790
820 IF U%(S1,S2)=0 THEN 790
830 IF TM>T(S1,S2) THEN 780
840 IF U%(S3,S4)<>1 THEN 790
860 IF ABS(S1-S3)+ABS(S2-S4)>3 THEN GOSUB 118:GOTO 790
880 T(S1,S2)=T(S1,S2)-TM:T(S3,S4)=T(S3,S4)+TM:RETURN
890 PRINT C$$S$FROM.TO":GOSUB 640:IF AB THEN 650
895 GOSUB 1630:IF E=1 THEN 890
900 IF S%(S1,S2)<1 THEN 890
920 IF U%(S3,S4)<>0 THEN 890
940 IF ABS(S1-S3)+ABS(S2-S4)>3 THEN GOSUB 118:GOTO 890
960 S%(S1,S2)=S%(S1,S2)-1:S%(S3,S4)=S%(S3,S4)+1
970 D%=INT(T(S1,S2)/(S%(S1,S2)+1))
980 T(S1,S2)=T(S1,S2)-D%:T(S3,S4)=T(S3,S4)+D%:RETURN
990 IF U%(X,Y)<>1 THEN GOSUB 130:GOTO 650
1010 IF T(X,Y)<1 THEN GOSUB 130:GOTO 650

```

```

1030 GOSUB 638:TA=R:N%=TA/9999:IF TA=0 OR AB THEN 650
1050 IF TA>T(X,Y) THEN GOSUB 130:GOTO 1030
1070 GOSUB 2000
1080 SK=INT(TA*RND(1)):T(X,Y)=T(X,Y)-SK:TK=TK+SK
1090 PRINT C$S$T(X,Y)"_LEFT":GOSUB 120
1100 Z%=1+INT(TA/8000)+INT(RND(1)*3):IF Z%>7 THEN Z%=7
1110 GOSUB 1770:RETURN
1120 IF M%<1 THEN PRINT S$"{rvs}NO_MISILES":GOSUB 120:GOTO 650
1130 M%=M%-1:POKE S,50:FOR Z=1 TO 99:NEXT:POKE S,0:
PRINT C$S$"LAUNCHED":GOSUB 120
1140 GOSUB 2000:NK=INT(N(X,Y)*RND(1)):TK=TK+NK:N(X,Y)=N(X,Y)-NK
1150 SK=INT(T(X,Y)*RND(1)):O%=SK/350
1160 T(X,Y)=T(X,Y)-SK:TK=TK+SK
1170 PRINT C$S$"CASUALTIES":PRINT S$"{down}"NK+SK:GOSUB 120
1180 IF U%(X,Y)=1 OR S%(X,Y)=0 THEN 1230
1190 IF T(X,Y)/S%(X,Y)>200 THEN 1230
1200 IF S%(X,Y)<0 THEN S%(X,Y)=0
1205 IF O%>S%(X,Y) THEN O%=S%(X,Y)
1210 S%(X,Y)=S%(X,Y)-O%
1220 PRINT C$S$O% "_SHIP(S)_SUNK":GOSUB 120
1230 Z%=RND(1)*9+1:GOSUB 1770:RETURN
1240 PRINT C$S$"FROM":GOSUB 640:IF AB THEN 650
1245 GOSUB 1630:IF E=1 THEN 1240
1250 IF W%(S1,S2)=0 THEN GOSUB 130:GOTO 1240
1270 IF ABS(X-S1)+ABS(Y-S2)>=7 THEN GOSUB 118:GOTO 1240
1290 GOSUB 638:NP%=R:N%=NP%:IF NP%=0 OR AB THEN 650
1320 IF NP%>W%(S1,S2) THEN PRINT S$"{rvs}TOO_MANY":GOSUB 120:
GOTO 1290
1330 GOSUB 2000:Z%=NP%*RND(1)/3+1:IF Z%>9 THEN Z%=9
1340 C%=NP%*RND(1):PRINT C$S$C% "_PLANES":PRINT S$"{down}DESTROYED":
GOSUB 120
1350 GOSUB 1770:W%(S1,S2)=W%(S1,S2)-C%:GOTO 510
1360 IF U%(X,Y)<>0 THEN GOSUB 130:GOTO 650
1380 N%=S%(X,Y)*20:IF N%<1 THEN PRINT S$"{rvs}NO_SHIPS":GOSUB 120:
GOTO 650
1400 IF T(X,Y)/S%(X,Y)>100 THEN 1420
1410 PRINT S$"{rvs}TOO_FEW_SAILORS":GOSUB 120:GOTO 670
1420 GOSUB 2000:Z%=S%(X,Y)+INT(RND(1)*4):IF Z%>7 THEN Z%=7
1430 GOSUB 1770:SK=INT(RND(1)*T(X,Y))
1435 O%=INT(SK/300):IF O%>S%(X,Y) THEN O%=S%(X,Y)
1440 S%(X,Y)=S%(X,Y)-O%
1450 IF S%(X,Y)=0 THEN SK=T(X,Y):T(X,Y)=0
1460 TK=TK+SK:T(X,Y)=T(X,Y)-SK:PRINT C$S$SK "_KILLED"
1470 PRINT S$"{down}"O% "_SHIP(S)_SUNK":GOSUB 120:RETURN
1480 PRINT C$S$"COORD":GOSUB 640:IF AB THEN 650
1485 GOSUB 1630:IF E=1 THEN 1480
1490 X1=X:Y1=Y:X=S1:Y=S2:GOSUB 570:X=X1:Y=Y1:GOTO 650
1500 IF H=0 THEN 1520
1510 PRINT C$S$"{rvs}ALREADY_USED":GOSUB 120:RETURN
1520 H=1:GOSUB 2000:FOR A=X-1 TO X+1:FOR B=Y-1 TO Y+1
1530 IF A<0 THEN 1610

```

```

1540 IF B<0 THEN 1600
1550 IF B>4 THEN 1610
1560 IF A>4 THEN 1620
1570 POKE P(A,B),86:T(A,B)=0:NK=INT(20*RND(1)*N(A,B))
1580 IF NK>N(A,B) THEN NK=N(A,B)
1590 N(A,B)=N(A,B)-NK:TK=TK+NK
1600 NEXT B
1610 NEXT A
1620 G%=86:GD=GD+INT(800+RND(1)*1500):RETURN
1630 E=0:GOSUB 2100:IF E THEN RETURN
1635 S1=VAL(LEFT$(R$,1)):S2=VAL(MID$(R$,2,1))
1640 S3=VAL(MID$(R$,3,1)):S4=VAL(RIGHT$(R$,1))
1650 IF S1<0 OR S1>4 OR S2<0 OR S2>4 THEN 1690
1660 IF LEN(R$)<3 THEN RETURN
1670 IF S3<0 OR S3>4 OR S4<0 OR S4>4 THEN 1690
1680 RETURN
1690 GOSUB 130:E=1:RETURN
1710 PRINT "NO";:DG=5:RETURN
1720 PRINT "LITTLE";:DG=20:RETURN
1730 PRINT "SOME";:DG=100:RETURN
1740 PRINT "MAJOR";:DG=150:RETURN
1750 PRINT "SEVERE";:DG=300:RETURN
1760 PRINT "GRAVE";:DG=400:RETURN
1770 PRINT C$S$;:ON Z%GOSUB 1710,1710,1720,1730,1730,1740,1750,1750,
    1760
1780 PRINT "_DAMAGE_TO":PRINT S$"{down}"G$:GOSUB 120
1790 GD=GD+DG:IF GD<K% THEN RETURN
1800 PRINT C$S$"{rvs}"G$"_IS_DEAD{off}":GOSUB 120
1810 PRINT C$S$"HE_KILLED":PRINT S$"{down}"INT(TK):GOSUB 120:
    GOSUB 120:GOTO 2040
1830 IF DG<200 OR DG>=400 THEN 1930
1840 GOSUB 340:GOSUB 570:PRINT C$S$G$"_ON":PRINT S$"{down}RAMPAGE":
    GOSUB 120
1850 IF U%(X,Y)=1 THEN 1900
1860 IF S%(X,Y)<1 THEN 1930
1870 SK=INT((RND(1)*T(X,Y))/S%(X,Y))
1880 T(X,Y)=T(X,Y)-SK:TK=TK+SK:S%(X,Y)=S%(X,Y)-1
1890 PRINT C$S$"SHIP_SUNK":PRINT S$"{down}KILLING_"SK:GOSUB 120:
    GOTO 1930
1900 NK=INT(N(X,Y)*RND(1)):SK=INT(T(X,Y)*RND(1))
1910 PRINT C$S$NK+SK"_KILLED":GOSUB 120:TK=TK+NK+SK
1920 N(X,Y)=N(X,Y)-NK:T(X,Y)=T(X,Y)-SK
1930 RETURN
1940 GOSUB 340:PRINT C$S$G$"_IS":PRINT S$"{down}IN_TOKYO!":GOSUB 120
1950 NK=INT(N(3,3)*RND(1)/3):SK=INT(T(3,3)*RND(1))
1960 N(3,3)=N(3,3)-NK:T(3,3)=T(3,3)-SK:TK=TK+SK+NK
1970 IF N(3,3)<1E4 THEN PRINT C$S$"{rvs}TOKYO_DESTROYED!":GOSUB 120:
    GOTO 2020
1990 PRINT C$S$NK+SK:PRINT S$"{down}KILLED":GOSUB 120:RETURN
2000 PO=P(X,Y):FOR I=1 TO 10+N%:POKE PO,214:POKE S,150:POKE S,0:
    POKE PO,86:POKE S,75

```

```

2010 N%=0:POKE S,0:NEXT:POKE PO,135:RETURN
2020 PRINT C$$ $ INT(TK):PRINT S$"{down}HAVE_DIED":GOSUB 120
2030 PRINT C$$ $G$"_WINS!!":GOSUB 120
2040 PRINT C$$ $"PLAY_AGAIN?":PRINT S$DF$;:INPUT A$
2060 IF LEFT$(A$,1)<>"N" THEN CLR:RR=1:GOTO 100
2070 POKE S+3,0:POKE S+2,0:POKE S,0:END
2100 IF R$="" THEN E=1:RETURN
2110 FOR Z=1 TO LEN(R$):Z$=MID$(R$,Z,1):IF Z$<"0" OR Z$>"9" THEN
    E=1:RETURN
2120 NEXT Z:RETURN

```



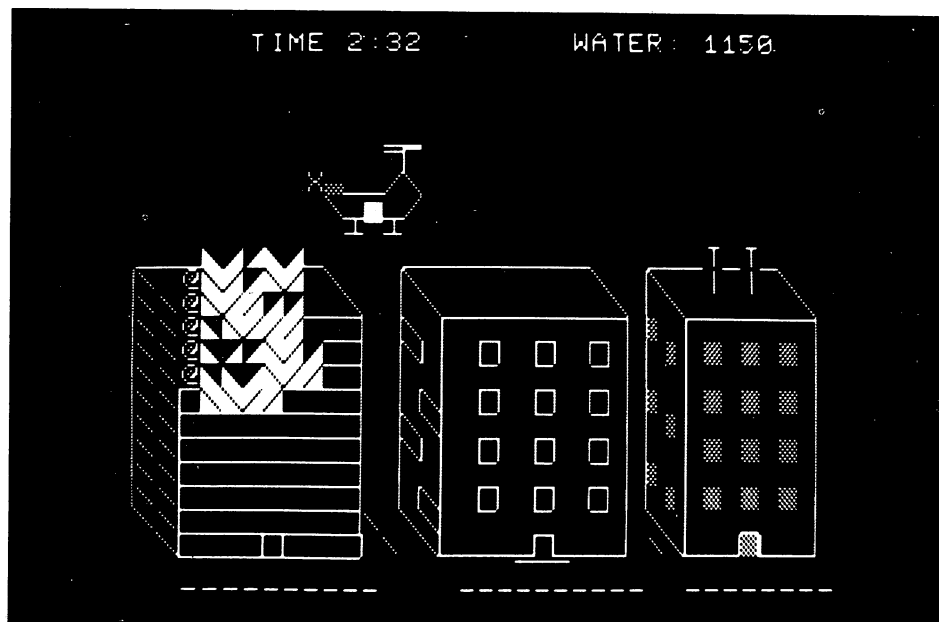
FIRE

By Brian Sawyer



There is a fire in a downtown high-rise building, and you have three minutes to put it out. Your fire-fighting helicopter carries 1500 gallons of water, which is enough to put out the fire if you can get to it quickly enough. The longer the fire burns, the more it spreads, and the more water it takes to put it out. If you run out of

water, you'll have to fly back to the water supply station at the right of the screen and refill. While you're off refilling the tank, the fire will continue to spread, so time is of the essence. You control your helicopter with the numeric keypad in the usual fashion as explained in the Introduction, and drop water by pressing [5].



```
0 CLR:POKE 59468,12
6 REM FIRE! COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/8/81 RJ
90 PG$="FIRE!":NM$="16":GOSUB 62000
100 Z=0:P=RND(-TI):READ LO,HI
110 FOR I=LO TO HI:READ X:POKE I,X:NEXT I
120 H$="{2°right}__{up}.__{2°left down $$}__{down 9°left}"
130 H$=H$+"_._._._.__{EO}__{down 8°left}"
140 H$=H$+"_X{($$NM)}__{down 8°left}"
150 H$=H$+"_.__{M$}_.__{N$}__{down 6°left}"
160 H$=H$+"_.__{1#1}_._"
170 PRINT "{clr 3°down}WANT_SOUND?_";SD=0:GOSUB 60000:
    IF LEFT$(IN$,1)="Y" THEN SD=16
175 IF IN$="" THEN PRINT TAB(12)"{up}N":FOR I=1 TO 500:NEXT
180 P=CRT+WD*13+4:E=29:V=3:TI$="000000":WT=1500:BP=P:CP=P:
    WL$="{3°left up rvs}_._":D$="4"
190 PRINT "{clr 8°down}"
200 PRINT "__{$$$$$$$$$}_._.__{$$$$$$$$$}_._.__{$$$}T{$}T{$}"
```

```

210 PRINT "{'}\_{.....M}\_{'M.....M}\_{'M}\_{[]}\_{[]}\_{M}"
220 PRINT "''\{\_{.....M}\_{'}\_{M.....M}'\}_{M}\_{.....{M}"
230 PRINT "''\{\O#####P}\_{'M}\_{O#####P'\}\_{O####P}"
240 PRINT "''\{\O#####P}\_{'M%}\_{O%}\_{O%'P'}\_{%\&}\_{&}\_{&'}"
250 PRINT "''\{\O#####P}\_{'}\_{%}\_{#}\_{#}\_{#'\}\_{%}\_{.....{'}"
260 PRINT "''\{\O#####P}\_{'M%}\_{O%}\_{O%'P'}\_{%\&}\_{&}\_{&'}"
270 PRINT "''\{\O#####P}\_{'MM%}\_{#}\_{#}\_{#'\}\_{%\}\_{.....{'}"
280 PRINT "''\{\O#####P}\_{'M%}\_{O%}\_{O%'P'}\_{%\&}\_{&}\_{&'}"
290 PRINT "''\{\O#####P}\_{'}\_{%}\_{#}\_{#}\_{#'\}\_{%\}\_{.....{'}"
300 PRINT "''\{\O#####P}\_{'M%}\_{O%}\_{O%'P'}\_{%\&}\_{&}\_{&'}"
310 PRINT "\_M\_{O#####P}\_{MM%}\_{#}\_{#}\_{#'\}\_{M}\_{%}\_{$}\_{$}\_{'}"
320 PRINT "\_{MO###OO###}\_{ML$$$$OL$$:_\_{ML$:&L$:}"
330 PRINT "\_{#####}\_{.....{EEE}"
340 PRINT "\_{-----}_\_{-----}_\_{-----}"
350 POKE P,233:FC=1
360 REM
370 IF FC=0 THEN 1000
380 IF TI>10800 THEN 1060
390 IF P=CRT+WD*11+10 OR P=CRT+WD*10+9 THEN P=P+6:G=10:GOTO 650
400 IF P=CRT+WD*11+15 OR P=CRT+WD*10+14 THEN P=P-7:GOTO 650
410 IF G=10 THEN G=2:GOTO 650
420 G=INT(RND(1)*3-1):IF G<>0 THEN 450
430 IF RND(1)>.5 THEN G=-WD:GOTO 450
440 IF RND(1)>.5 THEN G=WD
450 IF P<CRT+WD*10 OR P>CRT+WD*21+25 THEN 470
460 IF PEEK(P+G)<>32 AND PEEK(P+G+1)<>32 THEN 500
470 REM
480 IF PEEK(BP)>160 THEN P=BP:GOTO 500
490 IF PEEK(CP)>160 THEN P=CP
500 CP=P:P=P+G
510 GOSUB 970
520 IF I=6 AND E<34 THEN E=E+1
530 IF I=4 AND E>1 THEN E=E-1
540 IF I=2 AND V<4 THEN V=V+1
550 IF I=8 AND V>2 THEN V=V-1
560 PRINT "{home}"TAB(E)LEFT$(" {5°down}",V)H$;WL$;;IF I=8 THEN
PRINT "{2°down 2°left off}....";
570 PRINT :IF NOT (E=29 AND V=4) THEN 630
580 REM
590 IF WT>=1500 THEN 610
600 WT=1500:WL$="{3°left up rvs}_"
610 FOR Z=1 TO 300:NEXT:GOSUB 970
620 D$="4"
630 P9=P-CRT:V1=INT(P9/WD):H5=P9-V1*WD:IF H5<3 OR H5>23 THEN P=P-G
640 IF PEEK(P)=0 THEN P=P-G
650 F=INT(RND(1)*10):IF F<>5 THEN F=0
660 Z=PEEK(P):IF Z<>F THEN POKE P,233:IF Z<160 THEN FC=FC+1
670 Z=PEEK(P+1):IF Z<>F THEN POKE P+1,223:IF Z<160 THEN FC=FC+1
680 Z=PEEK(P+WD):IF Z<>F THEN POKE P+WD,206:IF Z<160 THEN FC=FC+1
690 Z=PEEK(P+WD+1):IF Z<>F THEN POKE P+WD+1,205:IF Z<160 THEN
FC=FC+1

```

```

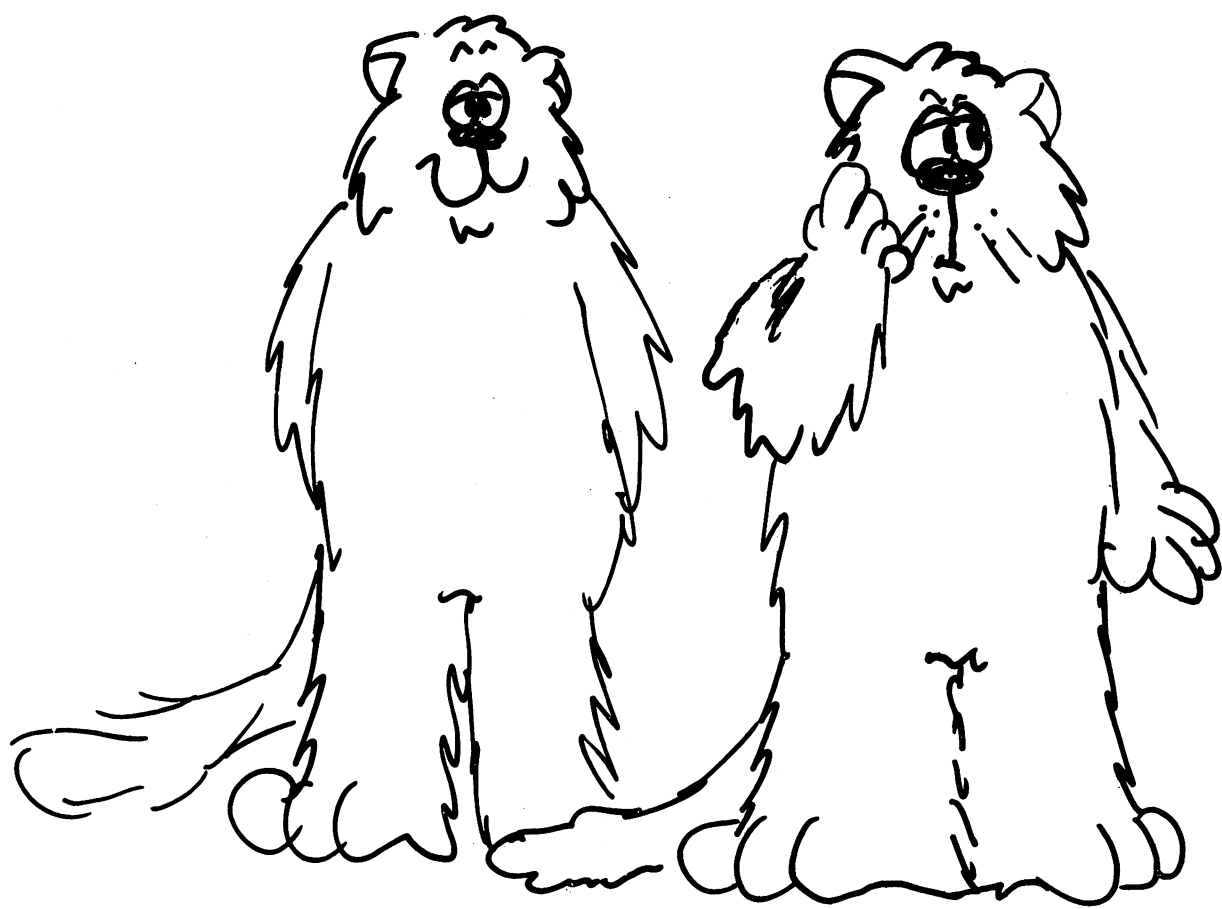
700 GOSUB 970
710 D1$=D$:GET D$:IF D$="" THEN D$=D1$
720 IF D$<>"5" THEN I=VAL(D$):GOTO 360
730 D$=D1$
740 REM
750 IF WT>0 THEN 810
760 POKE 59467,SD:IF SD THEN POKE 59466,15:POKE 59464,0
770 POKE 59464,50:POKE 59464,0
780 POKE 59464,50:POKE 59464,0
790 POKE 59464,50:POKE 59464,0
800 POKE 59467,0:POKE 59466,0:GOTO 360
810 WT=WT-50
820 POKE 59467,SD:IF SD THEN POKE 59466,15:POKE 59464,0
830 WL$="{3°left up}" + MID$("{off}_ {off $ off / off 9 off " rvs 8
    rvs 7 rvs # rvs}_",1+INT(WT/1501*9)*2,2)
840 S1=V*WD:POKE S1+CRT+WD*4+2+E,22
850 POKE 59464,81
860 FOR W=S1+CRT+WD*5+2+E TO CRT+WD*25 STEP WD
870 IF PEEK(W)=32 THEN POKE W,93:POKE W,32
880 IF PEEK(W+WD)<>32 THEN 900
890 NEXT
900 POKE 59467,0:POKE 59466,0:POKE 59464,0
910 IF PEEK(W)>160 THEN 940
920 W=W+WD:IF W>CRT+WD*25 THEN 360
930 GOTO 900
940 IF INT(RND(1)*5+1)=3 THEN POKE W,0:FC=FC-1:GOTO 360
950 POKE W,0:FC=FC-1:IF PEEK(W+WD)>160 THEN POKE W+WD,0:FC=FC-1
960 GOTO 360
970 PRINT "{home}";TAB(6);"{off}TIME_"MID$(TI$,4,1)": "RIGHT$(TI$,
    2)SPC(8);"WATER:"WT"{left}_
980 SYS 826:RETURN
990 REM
1000 TM$=TI$
1010 GET D$:IF D$<>" THEN 1010
1020 FOR N=1 TO 3000:NEXT
1030 PRINT "{clr}..YOU PUT THE FIRE OUT IN_"MID$(TM$,4,1);": ";
    RIGHT$(TM$,2);"! "
1040 XT=XT+1:GOTO 1070
1050 FOR N=1 TO 3000:NEXT
1060 PRINT "{clr}..YOU FAILED TO PUT OUT THE FIRE!"
1070 XN=XN+1:PRINT "{down
    2°right}YOU HAVE PUT OUT"XT"OUT OF"XN"FIRES."
1080 PRINT "{2°down 2°right}WANT TO PLAY AGAIN?_"":GOSUB 60000:
    IF LEFT$(IN$,1)<>"N" THEN 180
1090 POKE 59464,0:POKE 59466,0:POKE 59467,0
1100 END
1110 DATA 826,916
1120 DATA 160,255,132,1,160,132,132,2,160,0,169,0,162,0,177,1
1130 DATA 201,223,240,3,76,88,3,169,233,145,1,76,127,3,201,233
1140 DATA 240,3,76,102,3,169,223,145,1,76,127,3,201,205,240,3
1150 DATA 76,116,3,169,206,145,1,76,127,3,201,206,240,3,76,127

```

```

1160 DATA 3,169,205,145,1,198,1,240,3,76,72,3,198,2,169,255
1170 DATA 133,1,232,224,4,240,3,76,72,3,96
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60300 PRINT "{clr}":CLR:GOSUB 60430:GOTO 100
60430 CRT=32768:CR$=CHR$(13):WD=40:PRINT "{clr}"
60440 POKE CRT,96:IF PEEK(CRT+1024)<>96 THEN WD=80:PRINT CHR$(142)
60450 RETURN
60500 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$"..BY..BRIAN..SAWYER"
62010 PRINT "{down}.....CURSOR..#16..COPYRIGHT..(C)..1981"
62020 GOSUB 60500
62030 PRINT "{down}THE..ONE-PERSON..FIRE..DEPARTMENT"
62080 PRINT "{2°down}PRESS_{rvs}RETURN{off}..TO..BEGIN"
62090 GET T$:IF T$="" THEN 62090
62100 GOTO 60300

```

GAMES OF STRATEGY



DOTS

By Glen Fisher

This is a computerized form of a game that just about everyone has played at one time or another, in one form or another.

A grid of dots is drawn, each dot at the corner of a square. You and the PET take turns connecting the dots with lines. (Lines can go only along the edge of a square, and can't go through or past a dot.) When a square is boxed in by lines, the player who drew the last line gets the square, and marks it. The game is over when no more lines can be drawn, and the winner is the person with the most squares.

In this version, you "draw a line" by using the numeric pad to move a line-shaped cursor to indicate where you want your line, and then press

[RETURN]. Just remember that to change the line from horizontal to vertical (or vice versa), you must move it *diagonally* (by pressing [9], or [7], [1], or [3]). When it's not your turn, the PET will display a peacefully undulating line so you'll know you're not forgotten (and you'll accept your losses more calmly). If you want to stop the game, press [Q].

You can change the size of the board by changing the variables RS and CS at line 130. RS is the number of rows of squares, and CS is the number of columns of squares. There can't be more than 11 rows or 19 columns of squares. However, the bigger the board, the longer the PET takes to make its move.

```
0 CLR:POKE 59468,12
6 REM  COPYRIGHT (C) 1981 THE CODE WORKS
10 REM  AS OF 10/24/81 GF
90 PG$="DOTS":NM$="3":GOSUB 62000
100 CLR:GOSUB 60400
110 T=RND(-TI)
120 TBA=0:HZ=1:VT=2:HU=1:CO=2
130 RS=6:CS=6:SQ=RS*CS
140 RF=11-RS:CF=19-CS
200 R=RS+2:C=CS+2:DIM LC(R,C),VH(R,C),RT(4,4)
210 T=RND(-TI)
220 FOR I=0 TO 3:FOR J=0 TO 3:READ RT(I,J):NEXT J:NEXT I
230 DATA 3,4,2,6,4,5,1,7,2,1,0,8,6,7,8,9
240 DIM SV(1):SV(0)=0:SV(1)=1
250 Q$="{###EDC@FR$$$RF@CDE}":Q$=Q$+Q$:Q$=Q$+Q$
260 FOR I=1 TO 9:READ RM(I),CM(I):NEXT I
270 DATA 1,-1,2,0,1,1,0,-2,0,0,0,2,-1,-1,-2,0,-1,1
900 GOSUB 7000
910 PL=HU:SV(0)=0:SV(1)=1
990 IF PL=HU THEN GOSUB 1500:GOTO 1010
1000 GOSUB 5000:IF DIR=TBA THEN 2000
1010 GOSUB 5500:GOSUB 5600
1020 GOSUB 5800:IF N=0 THEN PL=3-PL
1030 IF SQ>0 THEN 990
1040 IF HQ>CQ THEN 1070
1050 IF HQ=CQ THEN 1080
1060 PRINT "{home}I,WON,WITH";CQ;"SQUARES_TO_YOUR";HQ;:GOTO 1100
1070 PRINT "{home}YOU,WON,WITH";HQ;"SQUARES_TO_MY";CQ;:GOTO 1100
```

```

1080 PRINT "{home}WE,TIED, WITH";HQ;"SQUARES EACH.";
1100 PRINT "{left}."
1105 PRINT ".....{up}"
1110 PRINT "WANT TO PLAY AGAIN?";:GOSUB 60000:T$=IN$
1120 IF LEFT$(T$,1)="Y" THEN 1200
1130 IF LEFT$(T$,1)<>"N" THEN PRINT "{up}";:GOTO 1105
1140 END
1200 FOR I=0 TO RS+2:FOR J=0 TO CS+2:LC(I,J)=0:VH(I,J)=0:NEXT J:
NEXT I
1210 SQ=RS*CS:CQ=0:HQ=0
1290 GOTO 900
1500 R=SV(0):C=SV(1)
1510 GOSUB 9000
1520 PK$=CHR$(PEEK(CRT+WD*(R+RF)+(C+CF))+128)
1530 GET T$:IF T$<>" " THEN 1600
1540 PRINT MID$("{GD}",1+(C AND 1),1);"{left}";
1550 PRINT PK$;"{left}";
1560 GOTO 1530
1600 IF T$<"1" OR T$>"9" THEN 1700
1610 T=VAL(T$):R=R+RM(T):C=C+CM(T)
1620 IF R<0 OR R>2*RS OR C<0 OR C>2*CS THEN R=R-RM(T):C=C-CM(T)
1700 IF T$<>CHR$(13) THEN 1800
1710 ROW=INT(R/2)+1:COL=INT(C/2)+1
1720 V=(R AND 1):H=1-V:DIR=HZ:IF V THEN DIR=VT
1730 IF VH(ROW,COL) AND (VT*V+HZ*H) THEN 1530
1740 SV(0)=R:SV(1)=C
1750 RETURN
1800 IF T$<>"Q" AND T$<>"_" THEN 1510
1810 PRINT "{home 22°down}";
1820 END
5000 MX=-1:DIR=TBA:ROW=COL=-1
5005 R=2*RS+1:C=0
5010 FOR I=1 TO RS
5020 FOR J=1 TO CS+1
5025 GOSUB 5200
5030 IF VH(I,J) AND VT THEN 5070
5040 F=RT(LC(I,J-1),LC(I,J))
5050 IF F<MX OR (F=MX AND RND(1)>.5) THEN 5070
5060 DIR=VT:MX=F:ROW=I:COL=J
5070 NEXT J
5080 NEXT I
5110 FOR I=1 TO RS+1
5120 FOR J=1 TO CS
5125 GOSUB 5200
5130 IF VH(I,J) AND HZ THEN 5170
5140 F=RT(LC(I-1,J),LC(I,J))
5150 IF F<MX OR (F=MX AND RND(1)>.5) THEN 5170
5160 DIR=HZ:MX=F:ROW=I:COL=J
5170 NEXT J
5180 NEXT I
5185 GOSUB 9000:PRINT ".....";

```

```

5190 RETURN
5200 GOSUB 9000:PRINT LEFT$(Q$,2*CS+1);
5210 Q$=MID$(Q$,2,255)+LEFT$(Q$,1)
5220 RETURN
5390 :
5400 TJ=TI+DL
5410 IF TI<TJ THEN 5410
5420 RETURN
5490 :
5500 LC(ROW,COL)=LC(ROW,COL)+1
5510 IF DIR<>HZ THEN 5540
5520 LC(ROW-1,COL)=LC(ROW-1,COL)+1
5530 VH(ROW,COL)=VH(ROW,COL) OR HZ:RETURN
5540 LC(ROW,COL-1)=LC(ROW,COL-1)+1
5550 VH(ROW,COL)=VH(ROW,COL) OR VT:RETURN
5590 :
5600 R=(ROW-1)*2:C=(COL-1)*2
5605 IF (R AND 1) OR (C AND 1) THEN STOP
5610 GOSUB 9000
5620 T$="{down }":IF DIR=HZ THEN T$="{right @}"
5630 PRINT T$;:IF PL=HU THEN RETURN
5640 DL=5:FOR QQ=0 TO 6
5650 GOSUB 5400:PRINT "{left}_";
5660 GOSUB 5400:PRINT "{left}";RIGHT$(T$,1);
5670 NEXT QQ:RETURN
5690 :
5700 R=R*2-1:C=C*2-1:GOSUB 9000
5710 SQ=SQ-1
5720 IF PL=HU THEN PRINT "{W}";:HQ=HQ+1:RETURN
5730 PRINT "{Q}";:CQ=CQ+1:RETURN
5790 :
5800 N=0
5810 R=ROW:C=COL:GOSUB 5900
5820 R=ROW-1:C=COL:GOSUB 5900
5830 R=ROW:C=COL-1:GOSUB 5900
5840 RETURN
5890 :
5900 IF LC(R,C)<>4 OR VH(R,C) AND 12 THEN RETURN
5910 VH(R,C)=VH(R,C) OR 12*PL:N=N+1
5920 GOSUB 5700:RETURN
5990 :
6000 PRINT "{clr}";
6010 FOR I=1 TO CS:PRINT "{2}_";:NEXT I:PRINT "{.}"
6020 PRINT "{up 0 down}"
6030 FOR J=2 TO RS
6040 FOR I=1 TO CS:PRINT "{[]}_";:NEXT I:PRINT "{3}"
6050 PRINT "{up + down}"
6060 NEXT J
6070 FOR I=1 TO CS:PRINT "{1}_";:NEXT I:PRINT "{=}"
6080 PRINT "{up - home}";
6090 RETURN

```

```

6900 END
7000 PRINT "{clr}":L$="{0}_":R$="{.}_":M$="{2}_"
7010 FOR I=2 TO RS:L$=L$+"{+}_":R$=R$+"{3}_":M$=M$+"{[]}_":NEXT I
7020 L$=L$+"{-}_":M$=M$+"{1}_":R$=R$+"{=}_"
7030 FOR R=0 TO 2*RS STEP 2
7040 C=0:GOSUB 9000:PRINT MID$(L$,R+1,2);
7050 FOR C=2 TO 2*CS-2 STEP 2:PRINT MID$(M$,R+1,2);:NEXT C
7060 PRINT MID$(R$,R+1,1)
7070 NEXT R
7080 RETURN
8990 :
9000 PRINT "{home}";:R=R+RF:C=C+CF
9010 IF R THEN PRINT LEFT$("{24°down}",R);
9020 IF C THEN PRINT LEFT$("{39°right}",C);
9030 R=R-RF:C=C-CF
9040 RETURN
50000 PRINT "{down}PRESS_{rvs}RETURN{off}_TO_CONTINUE:_"
50010 GET Z$:IF Z$="" THEN 50010
50020 RETURN
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>"" THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 GOSUB 60500:PRINT "{down}PRESS_{rvs}RETURN{off}_TO_";
60220 IF PN=LP THEN PRINT "BEGIN: ";:GOTO 60240
60230 PRINT "CONTINUE: ";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 100
60260 PN=PN+1:PRINT "{clr}";PG$; "_-_-CURSOR_#";NM$;
      "._.(C).1981....PAGE";PN; "{down}"
60270 GOTO 60500
60400 CRT=32768:CR$=CHR$(13):PRINT "{clr}":WD=40
60410 POKE CRT,96:IF PEEK(CRT+1024)=32 THEN WD=80:PRINT CHR$(142);:
      RETURN
60420 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$; "._.BY_GLEN_FISHER"

```

```

62005 PRINT "{down}.....CURSOR.#3{.}.COPYRIGHT.(C).1981"
62006 GOSUB 60500
62010 PRINT "{down}PLAYS.THE.ANCIENT.GAME.OF.'DOTS'."
62020 PRINT "{2°down}WANT.DIRECTIONS?.";:GOSUB 60000
62040 IF LEFT$(IN$,1)<>"Y" THEN 100
62100 LP=2:GOSUB 60200
62110 PRINT "PLAY.DOTS.AGAINST.THE.PET.-.YOU.EACH"
62120 PRINT "TRY.TO.CAPTURE.SQUARES.BY.COMPLETING"
62130 PRINT "THE.FOURTH.SIDE...NATURALLY,.YOU"
62140 PRINT "TRY.TO.AVOID.CREATING.THREE-SIDED"
62150 PRINT "SQUARES,.WHICH.THE.PET.WILL.CAPTURE."
62160 PRINT "{down}MOVE.THE.CURSOR.(USING.THE.NUMBER"
62170 PRINT "KEYS)..WHEN.THE.CURSOR.IS.WHERE.YOU"
62180 PRINT "WANT.TO.PUT.IT.PRESS.{rvs}RETURN{off}...THE.PET"
62190 PRINT "WILL.THEN.DECIDE.WHAT.MOVE.IT.SHOULD"
62210 PRINT "MAKE..DURING.THIS.TIME,.IT.WILL.DIS-"
62220 PRINT "PLAY.A.WAVY.LINE.SO.YOU'LL.KNOW.IT"
62230 PRINT "HASN'T.FORGOTTEN.YOU!"
62260 GOSUB 60200
62270 PRINT "THE.CURSOR.IS.CONTROLLED.BY.THESE"
62280 PRINT "NUMBER.CODES:{down}"
62290 PRINT ".4.-.MOVE.LEFT"
62300 PRINT ".6.-.MOVE.RIGHT"
62310 PRINT
62320 PRINT ".8.-.UP"
62330 PRINT ".2.-.DOWN"
62340 PRINT
62350 PRINT ".3.-.LEFT.TO.RIGHT.DIAGONAL.(DOWN)"
62360 PRINT ".9.-.LEFT.TO.RIGHT.DIAGONAL.(UP)"
62370 PRINT ".1.-.RIGHT.TO.LEFT.DIAGONAL.(DOWN)"
62380 PRINT ".7.-.RIGHT.TO.LEFT.DIAGONAL.(UP)"
62390 PRINT "{down}PRESS.{rvs}RETURN{off}.TO.ENTER.YOUR.MOVE."
62400 PRINT "{down}TO.STOP.THE.GAME,.PRESS.{rvs}Q{off up left $
down}.INSTEAD.OF"
62410 PRINT "ENTERING.YOUR.MOVE."
62430 GOSUB 60200

```




REVERSI

By Glen Fisher



Better known lately as “Othello,”¹ Reversi is an old game of strategy, invented by an Englishman in the 1800s. It is played on an 8×8 board with 64 pieces, white on one side, black on the other. The two colors show who owns a piece at any given time. When a piece is captured, it is turned over to show the color of the person taking it.

In REVERSI, each player takes turns placing a piece on the board. Any pieces already there may be captured, causing them to be flipped. Once a piece is placed, it is never moved again, although it may be flipped any number of times.

Captures are made by placing a piece in the same row, column, or diagonal as a piece you already own, with one or more of your opponent’s pieces separating your two. (No empty squares are allowed between your two — just enemy pieces.) The enemy pieces between your old piece and your newly placed piece (along rows, columns, or diagonals) are captured, and are all flipped to your color. (They will now all count as pieces you already own until your opponent takes them back.) Thus, you can capture pieces in up to eight directions at once.

Note that captures only occur when a piece is

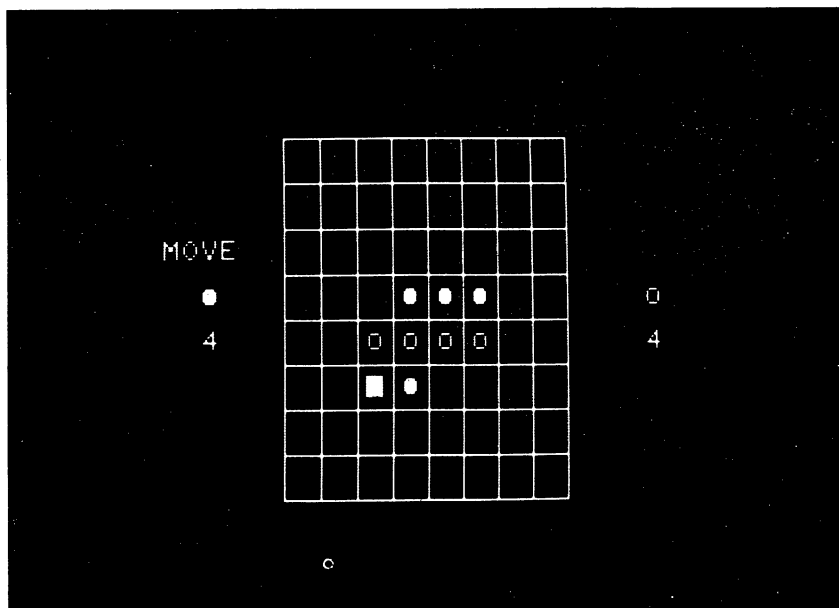
placed, and only in lines from the newly placed piece. If your opponent places a piece between two of yours, it stays his, even though you have a piece on both sides of it. Also, a captured piece cannot itself be used to make more captures in the same turn.

Basically, any move that captures one or more of your opponent’s pieces is a legal move. Conversely, any move that doesn’t capture anything is illegal. By the rules of capturing, this means that any legal move must place a piece next to one already on the board.

When you play the PET version of REVERSI, you first tell the PET whether you’re playing against it, or against another person. Next, you choose one of two patterns of the first four pieces on the board.

When it’s your turn to move, a cursor containing a piece of your color will appear on the board. Move it around using the numeric pad, and press [RETURN]. If the move is illegal, the cursor will reappear and you’ll have to move elsewhere. If the move is legal, all captures will be made, and your opponent will be given his turn. If you can’t make any legal moves, press [SHIFT-RETURN].

¹ “Othello” is a registered trademark of Gabriel Industries Inc.



```

0 CLR:POKE 59468,12
6 REM REVERSI! COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81 GF
90 PG$="REVERSI!":NM$="8":GOSUB 62000
100 DIM BD(9,9):WH=1:BL=2:MT=0
101 PRINT CHR$(142)
110 DIM RC(7),CC(7)
120 FOR I=0 TO 7:READ RC(I),CC(I):NEXT I
130 DATA -1,-1,-1,0,-1,1,0,1
132 DATA 1,1,1,0,1,-1,0,-1
140 DIM DC(9):FOR I=1 TO 9:READ DC(I):NEXT I
150 DATA 6,5,4,7,-1,3,0,1,2
240 PS$="_{QW}":PV$="_{WQ}"
250 CR$=CHR$(13)
500 DIM VL%(7,7),V(4)
510 V(1)=2:V(2)=3:V(3)=1:V(4)=4
520 FOR I=0 TO 7:V=V(ABS(3.5-I)+.5)/2
530 FOR J=0 TO 7:VL%(I,J)=VL%(I,J)+V:NEXT J
540 FOR J=0 TO 7:VL%(J,I)=VL%(J,I)+V:NEXT J
550 NEXT I
560 PRINT "HOW_MANY_PEOPLE_ARE_PLAYING?":PRINT TAB(9);"{#####
up}";TAB(29);
570 GOSUB 60000:IF IN$="" THEN PRINT TAB(29);"{up}1":FOR J=1 TO
500:NEXT
580 IF IN$="0" OR IN$="1" OR IN$="2" THEN NP=VAL(IN$):GOTO 600
590 IF IN$<>"" THEN 560
595 NP=1
600 S$=".....":T$="{lQlWl}"
605 PRINT "{clr}";S$;S$;"{0@2@.}";S$;"{0@2@.}"
610 PRINT S$;S$;T$;S$;T$
615 PRINT "START_WITH..(1){+@[3]_OR_(2){+@[3]_?"
620 PRINT S$;S$;T$;S$;"{lWlQl}"
625 PRINT S$;S$;"{-@l@=}" ;S$;"{-@l@= 2°up 3°right}";
630 GOSUB 60000:IF IN$="" THEN N=INT(RND(1)*2)+1:GOTO 700
640 IF IN$="1" OR IN$="2" THEN N=VAL(IN$):GOTO 700
650 PRINT "{2°down}PLEASE_ANSWER_'1'_OR_'2'." :GOTO 605
700 FOR I=1 TO 8:FOR J=1 TO 8:BD(I,J)=0:NEXT J,I
710 BD(4,4)=WH:BD(4,5)=BL
720 BD(5,4)=N:BD(5,5)=3-N
900 GOSUB 50000
910 FOR R=4 TO 5:FOR C=4 TO 5:GOSUB 39900
920 PRINT MID$(PS$,BD(R,C)+1,1):NEXT C:NEXT R
950 LR=3:HR=6:LC=3:HC=6
1000 M=5
1010 PRINT "{home 8°down}":PRINT "{down}"
1020 PRINT TAB(7);"{Q}";TAB(32);"{W}"
1030 WC=2:BC=2
1040 PRINT :PRINT TAB(6);WC;TAB(31);BC
2000 PRINT "{home 8°down}":CH=0
2010 T=(M AND 1):IF T THEN 2100
2020 PRINT TAB(30);"MOVE"

```

```

2025 PL=BL:PL$="{W}":SC$="{rvs Q off left}"
2030 ON NP+1 GOSUB 3500,3500,3000
2035 IF CH>0 THEN BC=BC+1
2040 GOTO 2200
2100 PRINT TAB(5);"MOVE"
2105 PL=WH:PL$="{Q}":SC$="{rvs W off left}"
2110 ON NP+1 GOSUB 3500,3000,3000
2120 IF CH>0 THEN WC=WC+1
2200 PRINT "{home 8°down}":PRINT TAB(5);:IF (M AND 1)=0 THEN
    PRINT TAB(30);:CH=-CH
2210 PRINT "....{3°down}"
2220 WC=WC+CH:BC=BC-CH
2230 PRINT TAB(6);WC;"{left}_" ;TAB(31);BC;"{left}_"
2240 IF CH=0 THEN SK=SK+1
2250 IF CH<>0 THEN SK=0
2260 M=M+1
2300 IF WC+BC=64 THEN 2500
2310 IF SK=2 THEN 2500
2320 IF WC=0 OR BC=0 THEN 2500
2330 IF CH=0 THEN 2000
2400 IF R<=LR AND LR>1 THEN LR=R-1
2410 IF R>=HR AND HR<8 THEN HR=R+1
2430 IF C<=LC AND LC>1 THEN LC=C-1
2440 IF C>=HC AND HC<8 THEN HC=C+1
2450 GOTO 2000
2500 PRINT "{home}THE GAME IS OVER."
2510 IF WC=BC THEN PRINT "{down}IT WAS A TIE GAME.":GOTO 2600
2520 PRINT "{home 8°down}":PRINT TAB(4);:IF BC>WC THEN
    PRINT TAB(29);
2530 PRINT "WINNER"
2600 PRINT "{home down}WANT TO PLAY AGAIN?_";:GOSUB 60000
2610 IN$=LEFT$(IN$,1):IF IN$<>"N" THEN 600
2620 END
3000 R=1:C=1:GOSUB 39900
3010 RV=1:TM=0
3100 T=ABS(BD(R,C))+1:PR$="{rvs}"+MID$(PV$,T,1)+"{off}"+MID$(PS$,T,1)
3110 RV=1:TM=0
3120 IF TI>TM THEN PRINT MID$(PR$,RV,2);"{left}";:RV=4-RV:TM=TI+30
3130 GET T$:IF T$="" THEN 3120
3140 PRINT RIGHT$(PR$,2);"{left}";
3145 IF T$="Q" THEN PRINT "{clr}":END
3150 IF T$>="1" AND T$<="9" THEN 3200
3160 IF T$=CHR$(13) THEN 3400
3170 IF T$=CHR$(141) THEN CH=0:RETURN
3180 GOTO 3120
3200 DR=DC(VAL(T$)):IF DR<0 THEN 3120
3220 R=R+RC(DR):C=C+CC(DR)
3230 IF R<1 THEN R=8
3240 IF R>8 THEN R=1
3250 IF C<1 THEN C=8
3260 IF C>8 THEN C=1

```

```

3270 GOSUB 39900:GOTO 3100
3400 IF BD(R,C)<>MT THEN 3120
3410 GOSUB 38200:IF CH=0 THEN GOSUB 39900:PRINT "{left}";
      BD(R,C)=MT:GOTO 3120
3420 RETURN
3500 CH=0:RM=0:CM=0:MC=-1:O=0
3510 FOR R=LR TO HR
3520 FOR C=LC TO HC
3530 IF BD(R,C)<>MT THEN 3610
3540 GOSUB 39900:PRINT SC$;
3550 GOSUB 38000
3560 IF CH>0 THEN CH=CH+VL$(R-1,C-1)
3570 IF CH=MC THEN O=O+1:IF RND(1)<1/O THEN RM=R:CM=C
3590 IF CH>MC THEN RM=R:CM=C:MC=CH
3600 PRINT "{left}";
3610 NEXT C
3620 NEXT R
3800 IF MC>0 THEN R=RM:C=CM:GOSUB 38200
3900 RETURN
38000 CH=0:FOR D=0 TO 7
38010 RC=RC(D):CC=CC(D)
38020 GOSUB 39000:IF N>0 THEN CH=CH+N
38030 NEXT D
38040 RETURN
38200 GOSUB 39900:BD(R,C)=PL:PRINT PL$,"{left}";
38210 CH=0:FOR D=0 TO 7
38220 RC=RC(D):CC=CC(D)
38230 GOSUB 39000:GOSUB 39100:IF N>0 THEN CH=CH+N
38240 GOSUB 39900:NEXT D
38250 RETURN
38300 GOSUB 39900:BD(R,C)=PL:PRINT PL$,"{left}";
38310 FOR D=0 TO 7
38320 RC=RC(D):CC=CC(D)
38330 N=OE(D):GOSUB 39400
38340 GOSUB 39900:NEXT D
38350 RETURN
39000 R0=R:C0=C:REM COUNT
39010 FOR I=1 TO 8
39020 R0=R0+RC:C0=C0+CC:BD=BD(R0,C0)
39030 IF BD=PL THEN N=I-1:RETURN
39040 IF BD=MT THEN 39070
39050 NEXT I
39070 N=-1:RETURN
39090 :
39100 R0=R:C0=C:IF N<1 THEN RETURN
39110 FOR I=1 TO N:REM FLIP
39120 R=R+RC:C=C+CC
39130 BD(R,C)=3-BD(R,C)
39132 GOSUB 39900:PRINT PL$
39135 FOR II=1 TO 3
39136 POKE 59467,16:POKE 59466,15:POKE 59464,16

```

```

39137 POKE 59464,0:POKE 59466,0:POKE 59467,0
39138 NEXT II
39140 NEXT I:R=R0:C=C0:RETURN
39190 :
39900 PRINT "{home}"
39910 PRINT LEFT$("{25°down}",2+R+R);
39920 PRINT LEFT$("{40°right}",10+C+C);
39930 RETURN
50000 PRINT "{clr 3°down}":TB=11
50010 PRINT TAB(TB);"{0@2@2@2@2@2@2@2@2@.}"
50020 T$="{[.]][.]][.]][.]][.]][.]][.]":PRINT TAB(TB);T$
50030 FOR I=1 TO 7:PRINT TAB(TB);"{+@[@[@[@[@[@[@[@3}":PRINT TAB(TB);
      T$:NEXT I
50040 PRINT TAB(TB);"{-@1@1@1@1@1@1@1@1@1@="}
50050 RETURN
60000 IN$="":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60260
60210 GOSUB 60500:PRINT "PRESS_{rvs}RETURN{off}_{TO}_{";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTINUE:"
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN PRINT "{clr}":GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"{-}_{CURSOR}_{#";NM$;"{(C)_{1981}_{PAGE}_{";
      PN
60270 GOSUB 60500:RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;"{_}_{BY}_{GLEN}_{FISHER}"
62005 PRINT "{down}_{CURSOR}_{#8}_{COPYRIGHT}_{(C)_{1981}"
62010 GOSUB 60500
62015 PRINT "{down}THE_{GAME}_{OF}_{REVERSI}"
62020 PRINT "{2°down}WANT_{DIRECTIONS?}_{";:GOSUB 60000
62030 IN$=LEFT$(IN$,1):IF IN$="Y" THEN 62100
62040 PRINT "{clr}":GOTO 100
62100 LP=1:GOSUB 60200
62110 PRINT "THE_{GAME}_{OF}_{REVERSI}_{DATES}_{BACK}_{TO}_{THE}"

```

```
62120 PRINT "1800'S IN ENGLAND..THE OBJECT IS TO"
62130 PRINT "OWN THE MOST PIECES AT THE END OF THE"
62140 PRINT "GAME."
62150 PRINT "{down}TO MOVE, USE THE NUMERIC PAD TO 7,8,9"
62160 PRINT "MOVE THE CURSOR AROUND...WHEN... \{\}/"
62170 PRINT "THE CURSOR IS WHERE YOU WANT TO 4{@ rvs}5{off @}6"
62180 PRINT "MOVE, PRESS {rvs}RETURN{off}..IF THE MOVE../{\}\\"
62190 PRINT "IS ILLEGAL, THE CURSOR WILL RE- 1,2,3"
62200 PRINT "APPEAR, AND YOU WILL HAVE TO"
62210 PRINT "MOVE SOMEWHERE ELSE... IF YOU HAVE NO"
62220 PRINT "MOVE AT ALL, PRESS{ rvs}SHIFT-RETURN{off}."
62230 PRINT "{down}WHILE THE MACHINE IS CHOOSING ITS MOVE,"
62240 PRINT "YOU WILL SEE IT SCANNING THE BOARD TO"
62250 PRINT "FIND ITS BEST MOVE."
62300 GOSUB 60200
```



PIEGRAM

By Glen Fisher

You and the PET are about to indulge in the classic prank of pie throwing. You and the PET take turns trying to hit each other with pies, or moving if the pies are getting too close for comfort. After each pie toss, you are given clues to help you find the PET.

The board is a 10×10 grid. When the game starts, you will be allowed to hide yourself on any of the squares. The PET is also on one of the squares, but you can't see it and it can't see you.

You are first given the choice of *throwing* or *moving*. If you want to throw a pie at the PET, press [T] (for throw). A blinking cursor will appear on the board, and the PET will ask you where you want to throw your pie. Using the numeric keypad, move the cursor to the square you want to hit and press [RETURN].

The pie will land in that square and the square will be assigned a number. On the side of the board, that same number will appear, and next to it will be a value representing how far away from the PET your pie landed. If you hit in the same row or column in which the PET is hiding, then the distance will be a whole number. If you do not hit in the same row or column, then the distance will be a fraction. If you are good with the

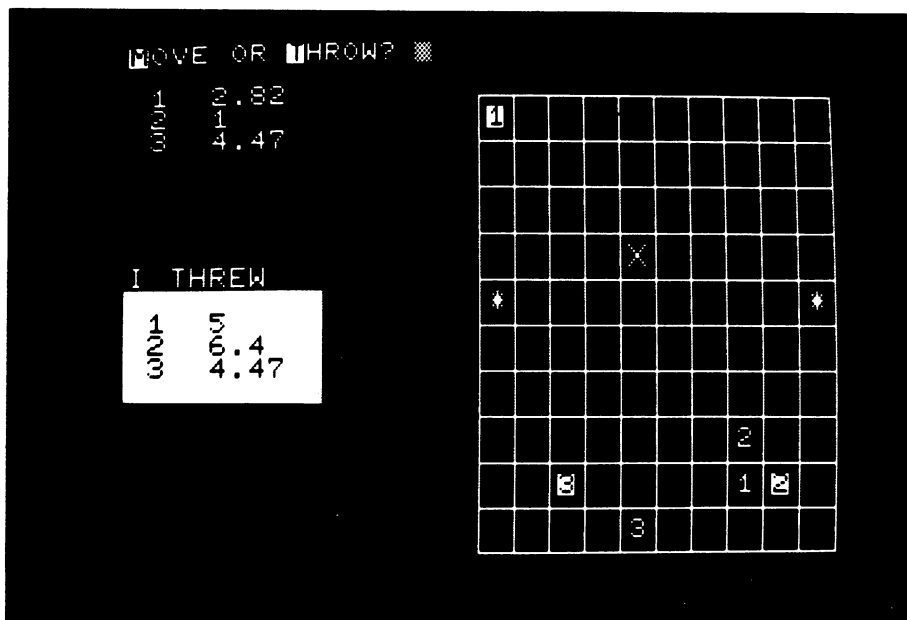
Pythagorean Theorem and square roots, then you can deduce which squares the PET could be hiding in.

If you don't want to be quite that precise (and it isn't necessary), you can just estimate. Using that estimate, when your next turn to throw comes up, choose one of the squares that you think the PET is in and throw a pie there. If you miss, then the new hint should further help you to narrow down where the PET is hiding.

If the PET is throwing its pies too close for comfort, you can move by pressing [M] (for move). A cursor will then appear. Move it to where you want to be, and press [RETURN]. If you move more than one square, the cursor will reappear, and you'll have to try somewhere else.

There are complications, however, because when you start getting too close to the PET, it will move to a new location. You must also be on the lookout for incoming pies, and move when they start getting too close. You can only move one square in any direction, and you cannot move into a square where a pie has landed because the floor is extremely slick there.

When you hit the PET, or are hit by a pie, the game is over.



```

0 CLR:POKE 59468,12
3 REM  BASED ON AN IDEA BY SHEILA DOLGOWICH
6 REM  PIEGRAM COPYRIGHT (C) 1981 THE CODE WORKS
10 REM  AS OF 10/24/81 TZ
90 PG$="PIEGRAM":NM$="12":GOSUB 62000
100 DIM B%(10,10),HS(4,3),CS(4,3)
110 PRINT CHR$(142)
120 FOR I=1 TO 9:READ RC(I),CC(I):NEXT
130 DATA 1,-1,1,0,1,1,0,-1,0,0,0,1,-1,-1,-1,0,-1,1
160 BL$="{off}....."
200 DN$="{home 10°down}"
500 FOR I=0 TO 4:FOR J=0 TO 3:HS(I,J)=0:CS(I,J)=0:NEXT J,I:HS=0:
    CS=0
510 FOR I=1 TO 10:FOR J=1 TO 10:B%(I,J)=0:NEXT J,I
600 GOSUB 9000:HR=0:HC=0:R=5:C=5
620 PRINT "{home}WHERE ARE YOU?":GOSUB 9100:IF C$="." THEN 620
630 HR=R:HC=C:GOSUB 9300:GOSUB 9400:PRINT RV$;SY$
650 PR=INT(RND(1)*10)+1:PC=INT(RND(1)*10)+1
1000 PRINT "{home}";BL$:PRINT "{home
    rvs}M{off}OVE OR {rvs}T{off}HROW?.";:GOSUB 60000
1010 IF IN$="" THEN 1000
1020 IN$=LEFT$(IN$,1):IF IN$="Q" THEN 3120
1025 IF IN$<>"M" AND IN$<>"T" THEN 1000
1030 T$="MOV":IF IN$="T" THEN T$="THROW"
1040 PRINT "{home}";BL$:PRINT "{home}WHERE ARE YOU.";T$;"ING TO?"
1050 R=5:C=5:IF IN$="M" THEN R=HR:C=HC
1060 SB=B%(HR,HC):B%(HR,HC)=2:GOSUB 9100:B%(HR,HC)=SB:IF C$="." THEN
    1000
1070 IF IN$="T" THEN 1200
1100 IF ABS(R-HR)>1 OR ABS(C-HC)>1 THEN 1050
1110 IF B%(R,C)<>0 THEN 1050
1120 SR=HR:SC=HC:HR=R:HC=C:R=SR:C=SC:GOSUB 9400:GOSUB 9300:
    PRINT RV$;SY$
1130 R=HR:C=HC:GOSUB 9400:GOSUB 9300:PRINT RV$;SY$
1140 FOR I=0 TO 2:CS(I,3)=CS(I,3)+5:NEXT I
1150 GOTO 2000
1200 IF B%(R,C)<>0 THEN 1050
1210 IF R=HR AND C=HC THEN 1050
1400 HS(3,0)=R:HS(3,1)=C:HS(3,2)=INT(100*SQR((R-PR)^2+(C-PC)^2))/100
1410 FOR I=0 TO 3:I1=I+1
1420 R=HS(I,0):C=HS(I,1):IF R>0 THEN B%(R,C)=B%(R,C)+1:GOSUB 9300:
    GOSUB 9400:PRINT SY$
1430 :FOR J=0 TO 3:HS(I,J)=HS(I1,J):NEXT J
1440 NEXT I:PRINT "{home down}"
1450 FOR I=2 TO 0 STEP -1
1460 IF HS(I,0)>0 THEN PRINT 3-I;HS(I,2);"{left}...."
1470 NEXT I
1480 IF HS(2,2)=0 THEN 3000
1490 HS=HS+1:GOTO 2000
2000 PRINT "{home}";BL$:PRINT DN$;". ...."
2010 R=PR-HS(2,0):C=PC-HS(2,1):D=R*R+C*C

```



```

2000 IF D<=2 THEN 2100
2030 IF RND(1)>D/100 THEN 2500
2100 R=INT(RND(1)*10)+1:C=INT(RND(1)*10)+1
2120 R=PR+SGN(R-PR):C=PC+SGN(C-PC)
2130 IF B%(R,C)<>0 THEN 2100
2135 PC=C:PR=R
2140 PRINT DN$;"I_MOVED"
2150 GOTO 1000
2500 IF CS>2 THEN 2600
2510 R=INT(RND(1)*10)+1:C=INT(RND(1)*10)+1
2520 IF B%(R,C)<>0 THEN 2510
2530 IF R=PR AND C=PC THEN 2510
2540 GOTO 2900
2600 MR=0:MC=0:MS=-1
2610 FOR I=1 TO 10:FOR J=1 TO 10:S=0
2620 IF B%(I,J)<>0 THEN 2850
2630 R=I:C=J:GOSUB 9300:PRINT "{rvs}?{off left}";
2640 FOR K=0 TO 2:R=CS(K,0):C=CS(K,1):D=CS(K,2):W=CS(K,3):L=D-W:
H=D+W
2650 I0=I-R:J0=J-C:T=I0*I0+J0*J0
2660 IF L<=T AND T<=H THEN S=S+1
2670 NEXT K
2700 IF S>MS THEN MR=I:MC=J:MS=S
2800 R=I:C=J:GOSUB 9400:PRINT RV$;SY$
2850 NEXT J,I
2860 R=MR:C=MC
2900 CS(3,0)=R:CS(3,1)=C:CS(3,2)=(R-HR)^2+(C-HC)^2:CS(3,3)=0
2910 FOR I=0 TO 3:I1=I+1:R=CS(I,0):C=CS(I,1)
2920 IF R>0 THEN B%(R,C)=B%(R,C)-1:GOSUB 9300:GOSUB 9400:PRINT RV$;
SY$
2930 FOR J=0 TO 3:CS(I,J)=CS(I1,J):NEXT J
2940 NEXT I:PRINT DN$;"I_THREW":PRINT "{rvs}....."
2950 FOR I=2 TO 0 STEP -1
2960 IF CS(I,0)>0 THEN PRINT "{rvs}";3-I;"{left}.....{7°left}";
INT(100*SQR(CS(I,2)))/100
2970 NEXT I:PRINT "{rvs}....."
2980 IF CS(2,2)=0 THEN 3000
2990 CS=CS+1:GOTO 1000
3000 W$="YOU":L$="ME":SH=0:IF B%(R,C)<0 THEN W$="I":L$="YOU":SH=1
3010 PRINT "{home}";BL$:PRINT "{home}";W$;"_GOT_";L$;"."
3020 GOSUB 9300:FOR I=1 TO 100:PRINT MID$("{rvs off}",(1 AND I)+1,
1);"*{left}";:FOR J=1 TO 10:NEXT J,I
3030 IF SH THEN R=PR:C=PC:GOSUB 9300:PRINT "{rvs V}":PRINT DN$;
"{rvs V off}_"_"ME_"
3100 PRINT "{home}";BL$:PRINT "{home}WANT_TO_PLAY_AGAIN?_";:
GOSUB 60000
3110 IF LEFT$(IN$,1)<>"N" THEN 500
3120 PRINT "{clr}":END
9000 R$="+@[@[@[@[@[@[@[@3}":
S$="{[]}_[]}_[]}_[]}_[]}_[]}_[]}_[]}_[]}_[]}_[]}_{}"
9010 BP=18

```



```

60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 QL=245:CR$=CHR$(13)
60410 IF PEEK(50000)=0 THEN RETURN
60420 QL=216
60430 RETURN
60500 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;"..BY..GLEN..FISHER"
62010 PRINT "{down}.....CURSOR.#12..COPYRIGHT.(C)..1981"
62020 GOSUB 60500
62030 PRINT "{down}A..PIE-THROWING..CONTEST"
62080 PRINT "{2°down}WANT..DIRECTIONS?..";:GOSUB 60000
62090 IF LEFT$(IN$,1)<>"Y" THEN 60300
62100 LP=4:GOSUB 60200
62110 PRINT "YOU..AND..THE..PET..ARE..ABOUT..TO..INDULGE..IN"
62120 PRINT "{down}THAT..GREAT..OLD..AMERICAN..PASTIME..OF..PIE-"
62130 PRINT "{down}THROWING..AS..MIGHT..BE..EXPECTED,..THERE"
62140 PRINT "{down}ARE..ONLY..ENOUGH..RULES..TO..PREVENT..INJURY"
62142 PRINT "{down}(WHAT..FEW..THERE..ARE,..ARE..REQUIRED..BY..TE{left
ins}H"
62144 PRINT "{down}OCCUPATIONAL..SAFETY..AND..HEALTH..ADMIN-"
62146 PRINT "{down}ISTRATION.)"
62148 GOSUB 60200
62150 PRINT "YOU..AND..THE..PET..TAKE..TURNS..EACH..TURN,"
62160 PRINT "{down}YOU..MAY..EITHER..MOVE..(TO..AVOID..A..THROWN"
62170 PRINT "{down}PIE)..OR..THROW..A..PIE..YOURSELF..YOU..CAN"
62180 PRINT "{down}THROW..A..PIE..INTO..ANY..EMPTY..SQUARE..YOU"
62190 PRINT "{down}WANT..TO.....{up ##### down}"
62220 PRINT "{2°down}YOU..CAN..MOVE..ONLY..ONE..SQUARE..(TO..AVOID"
62230 PRINT "{down}SLIPPING..ON..PIE..FILLING)..AND..YOU..CAN'T"
62240 PRINT "{down}MOVE..INTO..A..SQUARE..WITH..A..PIE..IN..IT..(TE{left
ins}H"
62250 PRINT "{down}FLOOR..IS..SLICK..THERE). "
62260 GOSUB 60200
62270 PRINT "EACH..TIME..YOU..ARE..ASKED..WHERE..YOU..WANT"
62280 PRINT "{down}TO..MOVE..OR..THROW..TO,..A..BLINKING..SQUARE"
62290 PRINT "{down}WILL..APPEAR..ON..THE..BOARD..YOU..MOVE..THE"
62300 PRINT "{down}SQUARE..TO..WHERE..YOU..WANT..TO..MOVE/THROW"
62310 PRINT "{down}TO,..AND..HIT..{rvs}RETURN{off}..IF..THE..CHOICE..IS"
62320 PRINT "{down}IMPROPER,..THE..SQUARE..WILL..REAPPEAR,..AND"
62330 PRINT "{down}YOU'LL..HAVE..TO..GO..SOMEWHERE..ELSE."
62340 GOSUB 60200
62350 PRINT "TO..MOVE..THE..SQUARE,..USE..THE..NUMERIC..PA.{left ins}D"
62360 PRINT
        "{down}AS..USUAL,..{rvs}8{off}..IS..UP,..{rvs}4{off}..IS..LEFT,.....7..8..9"
62365 PRINT TAB(34);"{M|N}"
62370 PRINT "{rvs}1{off}..IS..DOWN..AND..LEFT,..AND..SO..ON.....4{@
rvs}..{off @}6"
62375 PRINT TAB(34);"{N|M}"
62380 PRINT "PRESSING..THE..{rvs}SPACE{off}..KEY..WILL.....1..2..3"

```

62390 PRINT

"{down}RETURN_YOU_TO_THE_{rvs}M{off}OVE_OR_{rvs}T{off}HROW"

62400 PRINT "{down}QUESTION."

62410 GOSUB 60200



COPS

By Glen Fisher

The police have received an anonymous call informing them that the infamous Smith gang is in town. Naturally, they immediately send out a squad car to bring them in. The Smiths know this, because they always listen to the police radio. If they can avoid being caught for long enough, the squad car will be called back to police headquarters and the Smiths will be safe.

This two-person game sounds simple, and it is. But there are a few interesting rules. Traffic laws prohibit left turns in the city and the police, upholders of the law that they are, can only go straight or turn right. The Smiths, being hardened criminals, can (and will) turn anywhere they like. However, the police squad car, equipped with a siren to clear traffic, can move twice as fast as the Smith's car.

A map of the city will be shown on your screen, with the positions of the Smiths' car and the squad car indicated. The Smiths' car is marked by a "+" and the squad car is marked by an arrow pointing in the direction that it is moving (the

blinking symbol indicates whose turn it is to move). The two cars take turns moving, with the squad car moving twice the distance of the Smiths' car.

The squad car's movement is controlled by the numeric pad. The Smiths' car is controlled in a similar way, but with the leftmost letter keys [W], [A], [D], and [X]. The [W] moves the Smiths up, [A] moves them left, [D] moves them right, and [X] moves them down.

The outlaws are captured when the squad car comes within one block of the Smiths' car. The Smiths escape if they can evade capture for 15 moves.

A helpful hint if you are playing as the wrong side of the law is to keep the Smiths' car on the left side of the squad car, since it takes them three moves to make a left turn instead of only one.

This game was officially entitled *The Hamstrung Squad Car* when discussed in the "Mathematical Games" column of the February and March, 1967, issues of *Scientific American*.

```
0 CLR:POKE 59468,12
6 REM COPS COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81 RJ
90 PG$="COPS":NM$="14":GOSUB 62000
100 DIM UD(4),LR(4),OK$(4)
101 PRINT CHR$(142)
110 FOR I=1 TO 4:READ UD(I),LR(I):NEXT I
120 DATA -1,0,0,1,1,0,0,-1
130 FOR I=1 TO 4:READ OK$(I):NEXT I
140 DATA "86{2°-}", "{-}62{-}", "{2°-}24", "8{2°-}4"
150 AR$="↑>V<"
160 RS=25:CS=25:MT=15
400 BL$="....."
500 MG=5:MD=9:T=MG+MG:T1=MG+1:RJ=12
510 CR=INT(RND(1)*(RS-T))+T1:CC=INT(RND(1)*(CS-T))+T1
520 RR=INT(RND(1)*(RS-T))+T1:RC=INT(RND(1)*(CS-T))+T1
530 Z=SQR((RR-CR)^2+ABS(RC-CC)^2):IF Z<MD OR Z>RJ THEN 510
540 CD=1:CP$=MID$(AR$,CD,1):RB$="+"
600 EM$="":GOSUB 10000
610 R=CR:C=CC:GOSUB 10100:PRINT MID$(CP$,CD,1);"{home}"
620 R=RR:C=RC:GOSUB 10100:PRINT RB$;"{home}"
700 TN=MT
```

```

900 PRINT "{home}MOVES":PRINT "LEFT:":PRINT TN;"{left}_"
1000 R=CR:C=CC:CH$=OK$(CD):SY$=CP$:GOSUB 10200:IF CH=0 THEN 1500
1010 NR=CR+2*UD(CH):NC=CC+2*LR(CH)
1020 GOSUB 3000
1060 CR=NR:CC=NC:PRINT EM$;"{home}"
1070 CD=CH:CP$=MID$(AR$,CD,1)
1080 R=CR:C=CC:GOSUB 10100:PRINT CP$;"{home}"
1090 IF ABS(CR-RR)<2 AND ABS(CC-RC)<2 THEN 2000
1100 R=RR:C=RC:CH$="WDXA":SY$=RB$:GOSUB 10200:IF CH=0 THEN 1500
1110 NR=RR+UD(CH):NC=RC+LR(CH)
1120 GOSUB 3000
1160 RR=NR:RC=NC:PRINT EM$;"{home}"
1170 R=RR:C=RC:GOSUB 10100:PRINT RB$;"{home}"
1180 IF ABS(CR-RR)<2 AND ABS(CC-RC)<2 THEN 2000
1190 TN=TN-1:IF TN>0 THEN 900
1200 PRINT "{home}ESCAPE!":PRINT BL$:PRINT BL$
1210 RB=RB+1:GOTO 2100
1500 PRINT "{home}SIGH...":PRINT BL$:PRINT BL$
1510 FOR Z=1 TO 500:NEXT Z:GOTO 2100
2000 PRINT "{home}JUSTICE":PRINT "TRIUMPHS!":PRINT BL$
2010 CP=CP+1
2100 GM=GM+1:FOR Z=1 TO 500:NEXT Z
2110 GET T$:IF T$<>" " THEN 2110
2120 PRINT "{clr}";TAB(10)"IN"GM"GAME";MID$("S",2+(GM<>1))":{down}"
2130 PRINT "COPS_HAVE_CAUGHT_ROBBERS"CP"TIME"MID$("S",2+(CP<>1))"."
2140 PRINT "ROBBERS_HAVE_ESCAPED"RB"TIME"MID$("S",2+(RB<>1))"."
2200 PRINT "{2°down}WANT_TO_PLAY_AGAIN?_";
2210 GOSUB 60000:IF LEFT$(IN$,1)<>"N" THEN 500
2220 END
3000 IF NR<1 THEN NR=1
3010 IF NR>RS THEN NR=RS
3020 IF NC<1 THEN NC=1
3030 IF NC>CS THEN NC=CS
3040 RETURN
3100 IF NR<1 THEN NR=NR+RS
3110 IF NR>RS THEN NR=NR-RS
3120 IF NC<1 THEN NC=NC+CS
3130 IF NC>CS THEN NC=NC-CS
3140 RETURN
10000 L$="{right}":FOR I=1 TO CS:L$=L$+EM$:NEXT I
10020 PRINT "{clr}":TB=7:PRINT "{up}";
10030 FOR I=1 TO 24:PRINT TAB(TB);L$:NEXT I
10040 PRINT TAB(TB);L$;"{home}"
10060 RETURN
10100 PRINT "{home}":POKE QL,R:PRINT "{up}";TAB(TB+C);:RETURN
10110 REM PK=32768+80*R+2*C+TB-40:CH=PEEK(PK)
10200 Z=1:TM=0:GOSUB 10100
10210 GET Z$:IF Z$<>" " THEN 10250
10220 IF TI<TM THEN 10210
10230 PRINT MID$("{rvs off}",Z,1);SY$;"{left}";
10240 Z=3-Z:TM=TI+20:GOTO 10210

```

```

10250 IF Z$="_" THEN CH=0:GOTO 10290
10260 FOR Z0=1 TO LEN(CH$)
10270 IF Z$=MID$(CH$,Z0,1) THEN CH=Z0:GOTO 10290
10280 NEXT Z0:GOTO 10210
10290 PRINT "{off}";SY$;"{left}";:RETURN
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{TO}_";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$; "_-_{CURSOR}_{#}";NM$;
      "._.(C).1981._.PAGE";PN
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 QL=245:CR$=CHR$(13)
60410 IF PEEK(50000)=0 THEN RETURN
60420 QL=216
60430 RETURN
60500 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$; "._.BY._GLEN._FISHER"
62010 PRINT "{down}._.CURSOR_{#14}._.COPYRIGHT._.(C).1981"
62020 GOSUB 60500
62030 PRINT "{down}COPS._'N'._ROBBERS."
62080 PRINT "{2°down}WANT_{DIRECTIONS?}_";:GOSUB 60000
62090 IF LEFT$(IN$,1)<>"Y" THEN 60300
62100 LP=4:GOSUB 60200
62110 PRINT "THE._POLICE._HAVE._RECEIVED._AN._ANONYMOUS{down}"
62120 PRINT "CALL._THAT._THE._INFAMOUS._SMITH._GANG._IS._IN{down}"
62130 PRINT "TOWN._.NATURALLY,._THEY._SEND._OUT._A._SQUAD{down}"
62140 PRINT "CAR._TO._BRING._THEM._IN._.THE._SMITHS._KNOW{down}"
62150 PRINT "THIS,._SINCE._THEY._MAKE._IT._A._POINT._TO{down}"

```

```

62160 PRINT "LISTEN TO THE POLICE FREQUENCIES. THEY, {down}"
62170 PRINT "OF COURSE, WANT TO STAY FREE. IF THEY {down}"
62180 PRINT "MANAGE LONG ENOUGH, THE SQUAD CAR WILL {down}"
62190 PRINT "BE SENT OFF ON ANOTHER CALL, AND THE {down}"
62200 PRINT "SMITHS WILL BE SAFE. "
62210 GOSUB 60200
62240 PRINT "CITY TRAFFIC LAWS PROHIBIT LEFT TURNS {down}"
62250 PRINT "ANYWHERE, AND THE SQUAD CAR, UPHOLDER F {left
ins} O {down}"
62260 PRINT "THE LAW THAT IT IS, CAN ONLY GO STRAIGHT {left
ins} H {down}"
62270 PRINT "OR TURN RIGHT. THE SMITHS WILL GO ANY {down}"
62280 PRINT "WAY THEY LIKE, AS THEY ALREADY HAVE FAR {down}"
62290 PRINT "WORSE CRIMES ON THEIR RECORDS. THE SQUAD {left
ins} A {down}"
62300 PRINT "CAR, HOWEVER, CAN MOVE FASTER THAN THE {down}"
62330 PRINT "SMITHS' CAR, SINCE IT HAS A SIREN TO {down}"
62340 PRINT "CLEAR TRAFFIC OUT OF ITS WAY. "
62350 GOSUB 60200
62360 PRINT "A MAP OF THE CITY WILL BE SHOWN ON YOUR {down}"
62370 PRINT "SCREEN, WITH THE POSITIONS OF THE SMITHS {left
ins} H {down}"
62380 PRINT "AND THE SQUAD CAR MARKED. THE SMITHS AE {left
ins} R {down}"
62390 PRINT "MARKED BY A '+' . THE SQUAD CAR, SINCE T {left
ins} I {down}"
62400 PRINT "HAS TO OBEY THE LAW, IS MARKED WITH ONE {down}"
62410 PRINT "OF FOUR ARROWS (↑, >, V, <) DEPENDING ON {down}"
62420 PRINT "WHICH DIRECTION IT IS FACING. THE SQUAD {down}"
62430 PRINT "CAR AND THE SMITHS TAKE TURNS MOVING, {down}"
62440 PRINT "WITH THE SQUAD CAR MOVING FIRST. EACH {down}"
62450 PRINT "MOVE, THE SQUAD CAR MOVES TWO SPACES, {down}"
62460 PRINT "AND THE SMITHS MOVE ONE. "
62470 GOSUB 60200
62490 PRINT "WHEN IT IS THE SQUAD CAR'S TURN TO MOV, {left
ins} E {down}"
62500 PRINT "ITS ARROW WILL BLINK. IF IT IS THE {down}"
62510 PRINT "SMITHS' TURN TO MOVE, THEIR CAR WILL {down}"
62520 PRINT "BLINK. TO MOVE, THE SQUAD CAR USES THE {down}"
62530 PRINT "{rvs}2{off},{rvs}4{off},{rvs}6{off}, AND {rvs}8{off}
KEYS. THE SMITHS USE THE, {left ins}
{left rvs}X{off down}"
62540 PRINT "{rvs}A{off},{rvs}W{off}, AND {rvs}D{off}
KEYS. TO QUIT PRESS {rvs}SPACE{off}. {down}"
62545 PRINT "..... IF THE SQUAD CAR GETS {down}"
62550 PRINT ".....↑..... WITHIN ONE SPACE OF THE.....↑"
62555 PRINT "{rvs}W{off}"; TAB(36); "{rvs}8{off}"
62560 PRINT "<{rvs}A{right}D{off}> SMITHS, THE SMITHS ARE CAP-
<{rvs}4{right}6{off}>"
62565 PRINT "{rvs}X{off}"; TAB(36); "{rvs}2{off}"

```



```
62570 PRINT "...V...TURED..IF.THE.SMITHS.CAN.....V{down}"
62580 PRINT "EVADE.CAPTURE.FOR.15.TURNS,,THEY.ESCAP.{left ins}E"
62590 GOSUB 60200
```



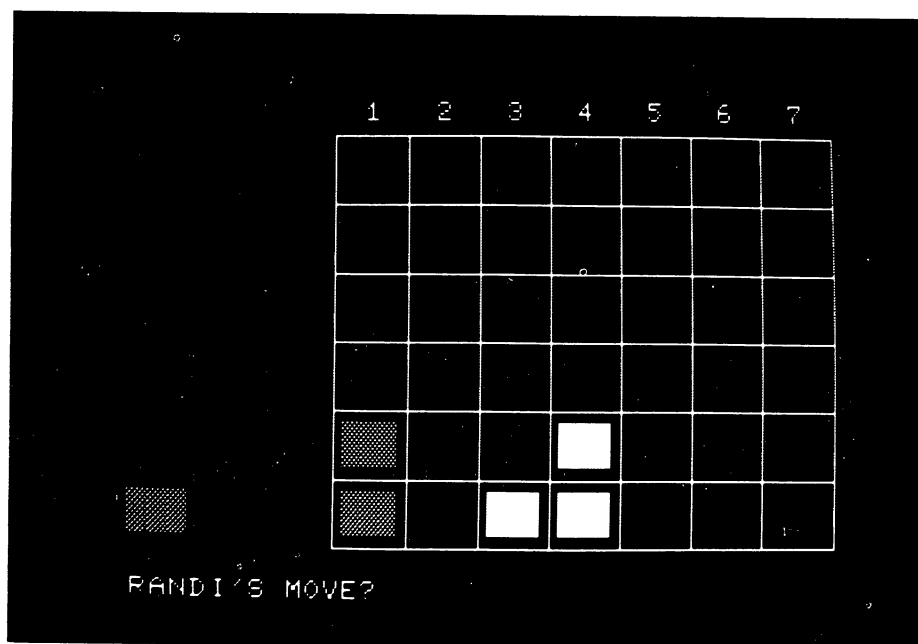
SPOT

By Art Carpet

SPOT is somewhat like tic-tac-toe, but a bit more difficult. You must get four of your squares in a line to win, but you can't just put your squares anywhere. When you select the column in which you want your square to go, it will drop down to the bottom of that column. Therefore, when you want your squares to be several rows up, you will have to build up some of

the columns. You will find that building up a column is tricky business, since the columns you stack may help you opponent to win.

The PET will tell you when it is your turn, and you may specify a column by pressing the number that corresponds to it. Your square will drop into that column and if it causes you to win, the PET will announce your victory.



```
0 CLR:POKE 59468,12
6 REM SPOT COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81 RJ
90 PG$="SPOT":NM$="17":GOSUB 62000
100 BL$=".....":GOSUB 1340
101 PRINT CHR$(142)
190 V=0:PRINT "{clr 2°down}.....1...2...3...4...5...6...7"
200 A$=".....{0@@@2@@@2@@@2@@@2@@@2@@@.}"
210 B$=".....{]...]}...]}...]}...]}...}"
220 C$=".....{+@@@[@@@[@@@[@@@[@@@[@@@[@@@3}"
230 D$=".....{-@@@1@@@1@@@1@@@1@@@1@@@1@@@=}"
240 PRINT A$:FOR T=1 TO 5:PRINT B$:PRINT B$:PRINT C$:NEXT:PRINT B$:
PRINT B$:PRINT D$
250 IF GA<>0 THEN 290
260 DIM A(7,6),B(42)
270 A$(1)="{&&& down 3°left &&&":A$(2)="{rvs}...{down
```

```

3°left}...{off}":A$(3)="...{down 3°left}..."
280 FOR T=1 TO 42:READ Z:B(T)=Z+32768:NEXT
290 GA=1:FOR T=1 TO 6:FOR Y=1 TO 7:V=V+1:A(Y,T)=B(V):NEXT:NEXT:V=0
300 X=INT(RND(TI)*2+1):AT=0
310 J=0:POKE QL,24:PRINT "{up}";BL$
320 GOSUB 2000
330 POKE QL,20:PRINT "{up}";A$(X);"{2°down}":PRINT N$(X);"'S_MOVE";
:
340 GOSUB 2100:IF R$="Q" THEN PRINT "{clr}":END
360 IF VAL(R$)<1 OR VAL(R$)>7 THEN 340
370 PRINT :POKE QL,24:PRINT "{up}";BL$:POKE QL,20:PRINT "{up}";
A$(3)
380 M=VAL(R$)
390 FOR T=1 TO 6
400 IF A(M,T)<9 THEN J=J+1
410 IF J=6 THEN 310
420 NEXT:J=0
430 AT=AT+1
440 FOR T=6 TO 1 STEP -1
450 IF A(M,T)>8 THEN 470
460 NEXT
470 V=M*4+7:H=T*3+2
480 FOR K=1 TO 2:TW=TI
490 POKE QL,H:PRINT "{up}"TAB(V)A$(X)
500 IF TI<TW+10 THEN 500
510 POKE QL,H:PRINT "{up}"TAB(V)A$(3)
520 IF TI<TW+20 THEN 520
530 NEXT:POKE QL,H:PRINT "{up}"TAB(V)A$(X)
540 A(M,T)=X↑3
550 REM CHECK FOR HORIZ WIN
560 FOR C=1 TO 4
570 IF A(C,T)+A(C+1,T)+A(C+2,T)+A(C+3,T)=4*(X↑3) THEN 590
580 NEXT:GOTO 640
590 M(1)=C:T(1)=T
600 M(2)=C+1:T(2)=T
610 M(3)=C+2:T(3)=T
620 M(4)=C+3:T(4)=T
630 GOTO 950
640 REM VERT WIN?
650 FOR C=1 TO 3
660 IF A(M,C)+A(M,C+1)+A(M,C+2)+A(M,C+3)=4*(X↑3) THEN 680
670 NEXT:GOTO 730
680 M(1)=M:T(1)=C
690 M(2)=M:T(2)=C+1
700 M(3)=M:T(3)=C+2
710 M(4)=M:T(4)=C+3
720 GOTO 950
730 REM L TO R DIAG?
740 FOR K=1 TO 4
750 FOR C=6 TO 4 STEP -1
760 IF A(K,C)+A(K+1,C-1)+A(K+2,C-2)+A(K+3,C-3)=4*(X↑3) THEN 780

```

```

770 NEXT:NEXT:GOTO 830
780 M(1)=K:T(1)=C
790 M(2)=K+1:T(2)=C-1
800 M(3)=K+2:T(3)=C-2
810 M(4)=K+3:T(4)=C-3
820 GOTO 950
830 REM CHECK FOR R TO L DIAG
840 FOR K=7 TO 4 STEP -1
850 FOR C=6 TO 4 STEP -1
860 IF A(K,C)+A(K-1,C-1)+A(K-2,C-2)+A(K-3,C-3)=4*(X↑3) THEN 880
870 NEXT:NEXT:GOTO 930
880 M(1)=K:T(1)=C
890 M(2)=K-1:T(2)=C-1
900 M(3)=K-2:T(3)=C-2
910 M(4)=K-3:T(4)=C-3
920 GOTO 950
930 IF AT=42 THEN 1036
940 X=3-X:GOTO 310
950 POKE QL,23:PRINT "{up}";TAB(10)N$(X)"_IS_THE_WINNER_!"
960 FOR I=1 TO 5
970 FOR Z=1 TO 4
980 M=M(Z):T=T(Z)
990 V=M*4+7:H=T*3+2
1000 POKE QL,H:PRINT "{up}"TAB(V)A$(3)
1010 FOR P=1 TO 100:NEXT
1020 POKE QL,H:PRINT "{up}"TAB(V)A$(X)
1030 NEXT:NEXT:GOTO 1040
1036 POKE QL,23:PRINT "{up}STALEMATE!_"
1040 POKE QL,24:PRINT "{up}WANT_TO_PLAY_AGAIN?_";:V=0:GOSUB 60000
1050 IF LEFT$(IN$,1)<>"N" THEN GA=1:PRINT "{clr}":GOTO 190
1060 END
1080 DATA 171,175,179,183,187,191,195,291,295,299,303,307,311,315,
411,415
1090 DATA 419,423,427,431,435,531,535,539,543,547,551,555,651,655,
659,663
1100 DATA 667,671,675,771,775,779,783,787,791,795
1340 PRINT "{clr 3°down}WHO'S_PLAYING_BLACK?_";:GOSUB 60000:
N$(1)=IN$
1345 IF IN$="" THEN N$(1)="PLAYER_1":PRINT TAB(21);"{up}";N$(1)
1350 PRINT "{2°down}WHO'S_PLAYING_WHITE?_";:GOSUB 60000:N$(2)=IN$
1355 IF IN$="" THEN N$(2)="PLAYER_2":PRINT TAB(21);"{up}";N$(2):
FOR I=1 TO 500:NEXT
1360 RETURN
2000 GET Z$:IF Z$<>"" THEN 2000
2010 RETURN
2100 ZT=TI:ZC=2
2110 GET R$:IF R$<>"" THEN 2140
2120 IF ZT<=TI THEN PRINT MID$("_?",ZC,1);"{left}";:ZC=3-ZC:ZT=TI+15
2130 GOTO 2110
2140 RETURN
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)

```

```

60010 GET Z$:IF Z$<>" THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
      ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{TO}_";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 60300
60260 PN=PN+1:PRINT "{clr}";PG$;"_{CURSOR}_#";NM$;
      "_{(C)}_{1981}_{PAGE}";PN
60265 GOSUB 60500
60270 RETURN
60300 PRINT "{clr}":CLR:GOSUB 60400:GOTO 100
60400 QL=245:CR$=CHR$(13)
60410 IF PEEK(50000)=0 THEN RETURN
60420 QL=216
60430 RETURN
60500 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;"_{BY}_ART_{CARPET}"
62010 PRINT "{down}_{CURSOR}_#17_{COPYRIGHT}_{(C)}_{1981}"
62020 GOSUB 60500
62030 PRINT "{down}FOUR_{IN}_A_{ROW}_TO_{WIN}..."
62080 PRINT "{2°down}WANT_{DIRECTIONS?}_";:GOSUB 60000
62090 IF LEFT$(IN$,1)<>"Y" THEN 60300
62100 LP=1:GOSUB 60200
62110 PRINT "THE_{OBJECT}_OF_{SPOT}_IS_{TO}_{GET}_{FOUR}_{PIECS}{left
ins}E{down}"
62120 PRINT "IN_{A}_{ROW},_{IN}_{ANY}_{DIRECTION}_{UNLIKE}_{TI}-{left
ins}C{down}"
62130 PRINT "TAC-TOE,_{THE}_{PIECES}_FALL_{TO}_{THE}_{BOTTOM}"
62140 PRINT "{down}OF_{THE}_{COLUMN}_IN_{WHICH}_{THEY}_{ARE}_{PUT}."
62150 PRINT "{2°down}NEEDLESS_{TO}_{SAY},_{YOUR}_{USUAL}_{TRICK}_{MOVES}"
62160 PRINT "{down}FROM_{TIC-TAC-TOE}_{WON}'T_{WORK}_{HERE}."
62170 GOSUB 60200

```



GAMES OF CHANCE



BJACK

By John Melissa

This program gives you a chance to play blackjack as it's played in the casinos of Las Vegas, with as much money as you care to risk. You are dealt two cards (face up) and the dealer also gets two (one face up, one face down). If you wish, you may be "hit" or dealt another card. You continue being "hit" until you choose to stop, or until the total value of your cards exceeds 21. If you pass 21, you're "busted," and you lose. Otherwise, the dealer reveals his hidden card and starts giving himself cards. He stops when his cards add up to 17 or greater. If he passes 21, he's busted and loses. If neither of you busts, the winner is the one with the higher total value.

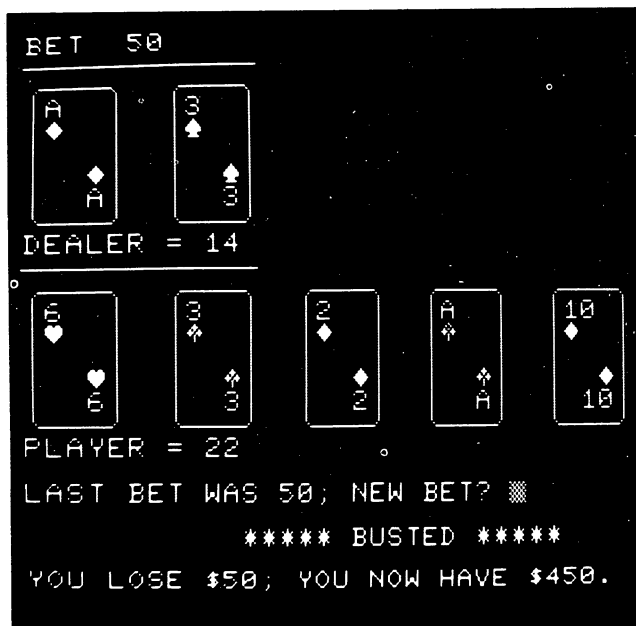
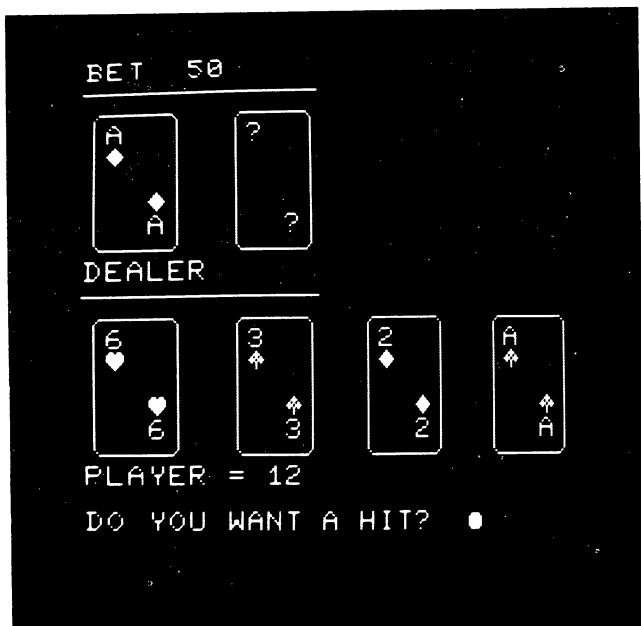
The values of the cards are the following: 2 through 10 = face value; court cards (Jack, Queen, King) = 10; Ace = either 1 or 11. The

PET counts Ace as 11 unless that would bust you.

If the first two cards you've been dealt add up to 21 (for instance, Ace plus a 10 or court card), you have blackjack, and win immediately. Likewise, if the dealer has blackjack, he wins.

If the dealer has an Ace showing, you may want to buy insurance against him having a blackjack. Insurance costs as much as you bet that hand. If he does have a blackjack, you get all your money back. If not, you lose the insurance money and the bet.

If your first two cards add up to 9, 10, or 11, you may "double down" if you like. To double down, the bet is doubled, and you get exactly one card, with no chance to be hit. After you get your card, the dealer gives himself as many as he likes, and things proceed as in a normal hand.



```
0 CLR:POKE 59468,12
6 REM BJACK COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81 RJ
80 CR$=CHR$(13)
90 PG$="BJACK":NM$="6":GOSUB 62000
100 DIM X(13),Y(13),D(52),B$(5),A$(14)
105 PRINT CHR$(142)
110 FOR I=1 TO 5:READ B$(I):NEXT I
```



```

115 DATA "{Z}","{A}","{S}","{X}","_"
120 FOR I=1 TO 14:READ A$(I):NEXT I
125 DATA A,2,3,4,5,6,7,8,9,10,J,Q,K,?
130 CR$=CHR$(13):BL$="....."
140 DN$="{home 21°down}"
150 PR$="HOW_MUCH_IN_CHIPS?_":IN=500:GOSUB 9600:B2=IN
160 B1=B1+B2
170 IF B>0 THEN 240
175 B=INT(.1*B2)
180 LB=B:GOTO 220
200 B1=B1+B:PRINT DN$;"{3°down}YOU_";MID$("LOSEWIN",-4*(B>0)+1,4);
205 PRINT "{_}$";MID$(STR$(B),2);";_YOU_NOW_HAVE_$";MID$(STR$(B1),
2);".{home}"
210 FOR J=1 TO 500:NEXT:IF B1<.01 THEN 9000
220 PRINT DN$;"{2°up}":PR$="LAST_BET_WAS"+STR$(LB)+";_NEW_BET?{_"
240 IN=LB:GOSUB 9600:B=IN:PRINT "{clr}BET_";B
250 IF B<.25 THEN 9100
260 IF B<=B1 THEN 300
270 PRINT "{clr down}YOU_ONLY_HAVE"B1
280 PRINT :PRINT "GET_MORE_CHIPS!{down}"
290 CLR:GOTO 100
300 LB=B
310 F1=0:F2=0
320 FOR J=1 TO 52
330 LET D(J)=1
340 NEXT J
400 X(13)=14
410 Y(13)=5
500 REM
505 GOSUB 1000
510 X(1)=C1
515 Y(1)=C2
516 IF C1>10 THEN C1=10
517 M=C1
520 GOSUB 1000
530 X(7)=C1
535 Y(7)=C2
536 IF C1>10 THEN C1=10
537 N=C1
540 GOSUB 1000
550 X(2)=C1
560 Y(2)=C2
561 IF C1>10 THEN C1=10
563 M=M+C1
565 GOSUB 1000
570 X(8)=C1
575 Y(8)=C2
576 IF C1>10 THEN C1=10
577 N=N+C1
580 PRINT "{@@@@@@@@@@@@@}"
600 J=7:P=1

```

```

605 GOSUB 2000
610 J=13:P=2
615 GOSUB 2000
620 PRINT "{6°down}":PRINT "DEALER"
670 PRINT "{@@@@@@@@@@@@}"
700 J=1:P=1:GOSUB 2000
720 J=2:P=2:GOSUB 2000
730 PRINT "{6°down}":PRINT "PLAYER"
735 M1=0
740 IF X(1)=1 OR X(2)=1 THEN M1=1
750 N1=0
755 GOSUB 2200:F1=1
760 IF X(7)=1 OR X(8)=1 THEN N1=1
770 I9=0:F1=1
775 REM
780 IF X(7)=1 THEN GOSUB 8000
790 IF N=11 AND N1=1 THEN 3000
800 IF I9=0 THEN 810
805 PRINT DN$;"{down}INSURANCE_LOSES_$";MID$(STR$(B/2),2);".":
      B1=B1-B/2
810 J=2
815 IF M=11 AND M1=1 THEN 5500
817 IF M>8 AND M<12 THEN GOSUB 7000:IF YN THEN 5000
820 IF M>20 THEN 5000
830 IF M=11 AND M1=1 THEN 5000
840 PRINT DN$
850 PRINT "{2°up}DO_YOU_WANT_A_HIT?.....{9°left}";
860 YN=1:GOSUB 9500
870 IF YN THEN GOSUB 4000:GOTO 820
880 GOTO 5000
890 GOTO 840
999 GOTO 9000
1000 C=INT(RND(1)*52+1)
1010 IF D(C)<>1 THEN 1000
1015 D(C)=0
1020 C1=INT(C/4)
1030 C2=C-C1*4
1032 C1=C1+1
1035 IF C2<1 THEN C2=C2+4:C1=C1-1
1099 RETURN
1900 PRINT "{home down}":REM FALLS INTO...
2000 REM
2020 PRINT TAB(P*7-7);
2030 PRINT "{U@@@I}";
2050 PRINT "{down left | down left | down left | down left | left
      down | left down K 2°left @ 2°left @ 2°left @ 2°left J left up
      | left up | left up | left up | left up |}";
2120 PRINT A$(X(J));"{down left}";
2125 IF X(J)=10 THEN PRINT "{left}";
2130 PRINT B$(Y(J));
2150 PRINT "{right 2°down}";B$(Y(J));"{left down}";

```

```

2155 IF LEN(A$(X(J)))>1 THEN PRINT "{left}";
2160 PRINT A$(X(J));"{5°up}";
2190 IF F1=0 THEN RETURN
2200 REM
2210 M2=M
2220 IF M1=1 AND M<12 THEN M2=M+10
2230 PRINT "{home 18°down 7°right}=";M2
2300 REM
2310 IF F2=0 THEN RETURN
2320 N2=N
2330 IF N1=1 AND N<12 THEN N2=N+10
2340 PRINT "{home 9°down 6°right}=";N2
2990 RETURN
3000 P=2:J=8
3040 GOSUB 1900
3042 IF I9=1 THEN 3100
3045 IF M=11 AND M1=1 THEN 5600
3050 PRINT DN$;"{down}DEALER_HAS_BLACKJACK"
3060 B=-B:GOTO 200
3100 N=0:IF M=11 AND M1=1 THEN 5620
3120 PRINT DN$;
      "{down}DEALER_HAS_BLACKJACK_--_LOSS_COVERED_BY_INSURANCE"
3130 GOTO 210
4000 J=J+1
4005 IF J>6 THEN 6000
4010 GOSUB 1000
4015 IF C1>10 THEN C1=10
4017 IF C1=1 THEN M1=1
4020 M=M+C1
4030 PRINT "{home 11°down}";
4040 P=J
4060 X(J)=C1
4070 Y(J)=C2
4090 GOSUB 2000
4400 RETURN
5000 J=8:P=2:F2=1
5020 GOSUB 1900
5030 IF M>21 THEN 5400
5090 J=J+1
5100 IF N>16 THEN 5600
5110 IF N<12 AND N1=1 AND N+10>17 THEN 5600
5120 GOSUB 1000
5130 X(J)=C1
5140 Y(J)=C2
5150 IF C1>10 THEN C1=10
5155 IF C1=1 THEN N1=1
5160 N=N+C1
5170 P=J-6
5180 GOSUB 1900
5186 FOR I=1 TO 400:NEXT
5190 GOTO 5090

```

```

5400 PRINT DN$
5410 PRINT ".....*****_BUSTED_*****"
5420 B=-B:GOTO 200
5500 J=8:P=2
5530 GOSUB 1900
5560 PRINT DN$
5570 PRINT ".....BLACKJACK!!"
5580 B=B*1.5:GOTO 200
5600 IF M<12 THEN IF M1=1 THEN M=M+10
5610 IF N<12 THEN IF N1=1 THEN N=N+10
5620 PRINT DN$
5625 IF N>21 THEN 5800
5630 IF M>N THEN 5700
5640 IF M<N THEN B=-B:GOTO 5700
5650 PRINT ".....STAND-OFF"
5660 PRINT "{down}CHIP_BALANCE_REMAINS_AT_$";MID$(STR$(B1),2);
      ".{home}"
5699 GOTO 210
5700 PRINT DN$;"{down}DEALER_HAS";N;"{left}";_YOU_HAVE";M;"{left}."
5710 GOTO 200
5800 PRINT DN$;"{down}***{_}DEALER_BUSTED{_}***":GOTO 200
6000 PRINT DN$:PRINT "SIX_CARDS_LESS_THAN_21_IS_A_WINNER!"
6020 GOTO 200
7000 PRINT DN$
7010 PRINT "{2°up rvs}DO_YOU_WANT_TO_DOUBLE?{off}_"";
7020 GOSUB 9400:IF YN=0 THEN RETURN
7100 B=B*2
7110 PRINT "{home}BET_";B
7120 GOSUB 4000
7130 YN=1:RETURN
8000 PRINT DN$;"{up rvs}WANT_INSURANCE?{off}_"";
8010 GOSUB 9400:IF YN THEN I9=1
8020 RETURN
9000 PRINT DN$
9005 PRINT "YOU_ARE_BROKE....."
9006 PRINT ".....{EEEEEE}....."
9010 PRINT "WOULD_YOU_LIKE_TO_PLAY_AGAIN?_";:GOSUB 60000
9020 IF LEFT$(IN$,1)="N" THEN 9100
9037 B=0
9040 PRINT "{clr}":GOTO 130
9100 PRINT "{clr}HOPE_YOU_ENJOYED_THE_GAME."
9110 PRINT "{down}GOODBYE."
9120 END
9400 YN=0
9500 ZC=2:ZM=0
9510 GET Z$:IF Z$<>" " THEN 9550
9520 IF TI<ZM THEN 9510
9530 PRINT MID$("_{Q}",ZC,1);"{left}";:ZC=3-ZC
9540 ZM=TI+30:GOTO 9510
9550 IF Z$<>"Y" AND Z$<>"N" AND Z$<>CR$ THEN 9520
9560 IF Z$<>CR$ THEN YN=Z$="Y"

```

```

9570 PRINT MID$("YN",2+(YN<>0),1)
9580 RETURN
9600 PRINT BL$:PRINT "{up}";
9605 CR$="":PRINT PR$;:GOSUB 60000:IF IN$="" THEN PRINT "{left}";IN:
GOTO 9640
9610 PRINT :FOR Z=1 TO LEN(IN$):Z$=MID$(IN$,Z,1)
9620 IF Z$<"0" OR Z$>"9" THEN 9650
9630 NEXT Z:IN=VAL(IN$)
9640 CR$=CHR$(13):RETURN
9650 PRINT "NO_";Z$;"'S_PLEASE."
9660 FOR Z=1 TO 2000:NEXT Z
9670 PRINT "{up}";BL$:PRINT "{2°up}";
9680 GOTO 9600
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 PRINT "{down}PRESS_{rvs}RETURN{off}_TO_";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN CLR:PRINT "{clr}";:GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_CURSOR_#";NM$;
"_(C)_1981_...PAGE";PN;"{down}"
60270 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$"_BY_JOHN_MELISSA"
62005 PRINT "{down}_CURSOR_#6_COPYRIGHT_(C)_1981"
62006 GOSUB 60500
62010 PRINT "{down}LAS_VEGAS-STYLE_BLACKJACK"
62020 PRINT "{2°down}WANT_DIRECTIONS?_";:GOSUB 60000
62030 IF LEFT$(IN$,1)<>"Y" THEN PRINT "{clr}";:RETURN
62040 LP=1:GOSUB 60200
62050 GOSUB 60500
62120 PRINT "FIRST,{_}YOU_MUST_ENTER_HOW_MANY_CHIPS"
62130 PRINT "YOU_WANT.{_}EACH_HAND,_YOU_WILL_BE_ASKED"
62140 PRINT "TO_MAKE_A_BET.{_}BET_ZERO_TO_END_THE"

```

```

62150 PRINT "GAME..IF.YOU.WISH.TO.PLACE.THE.SAME.BET"
62160 PRINT "AS.THE.LAST.HAND,.YOU.MAY.JUST.PRESS"
62170 PRINT "{rvs}RETURN{off}..WHEN.APPROPRIATE,.YOU.WILL.BE"
62180 PRINT "ASKED..IF..YOU..WANT..TO..DOUBLE..DOWN..OR..BUY"
62190 PRINT "INSURANCE."
62230 PRINT "{down}YOU.MAY.NOT.BET.MORE.THAN.YOU.HAVE"
62240 PRINT "IN.CHIPS.EXCEPT.TO.DOUBLE{.}DOWN."
62270 PRINT "{down}DOUBLE:.YOU.DOUBLE.YOUR.BET.AND.GET.ONE"
62280 PRINT "....(AND.ONLY.ONE).CARD."
62290 PRINT "{down}INSURANCE:.COSTS.HALF.YOUR.BET.AND.PAYS"
62300 PRINT "....TWO.TO.ONE.IF.DEALER.HAS.BLACKJACK."
62305 GOSUB 60500
62310 GOSUB 60200

```



YAHTZEE

By Glen Fisher

The object of Yahtzee¹ is to accumulate the highest number of points by rolling dice in specific combinations. Each turn consists of as many as three rolls of the dice: an initial roll of all five dice, and up to two rolls in which you may reroll some or all of the dice.

After each turn, you score points in one of 13 categories described below. Once you put a score in a category you can't change it later. For example, if you put a score of 6 under "sixes," and later get 24 points that you could have counted as sixes, you're stuck with the 6 you put there originally.

There is one exception to the "you catch it, you keep it" rule. It comes in when you roll more than one Yahtzee. If you have already scored 50 for a Yahtzee, and then roll another, you may choose to add 100 to the score for Yahtzees. Roll a third, and you can add another 100, and so on. If you cheat and roll 13 Yahtzees, you can get 1250 points for Yahtzees (50 for the first and 1200 for the other 12). Note that it applies only if you scored something. If you've zeroed that category, all the Yahtzees in the world will do you no good because that zero stays.

The following combinations will give you points:

- *Aces - sixes.* These are six different categories, but are treated very nearly the same. Each category corresponds to one of the numbers one through six, and a score in one of the categories is obtained by adding up only the occurrences of that number. For example, if after your final roll, your dice show 3-3-4-2-3, then your score in the threes' category would be 9 since you have three 3s ($3 + 3 + 3 = 9$). At the end of the game, if your scores in the aces through sixes categories add up to 63 or more, you get a bonus of 35 points. If you can roll three dice in each category (three aces, three twos, and so on), you'll get the bonus. However, since the bonus is based only on the total, you could take four sixes and balance out a bad roll of two fives.

- *Three-of-a-kind.* You must roll at least three of any one number to win points in this category, and your score is the sum of all five dice. Thus, if

you rolled 3-3-2-5-3, your score would be 16 ($3 + 3 + 2 + 5 + 3 = 16$). You can take a score in three-of-a-kind even if you have more than three of one number. For example, you could take 4-4-1-4-4 in three-of-a-kind, for 17.

- *Four-of-a-kind.* You must roll at least four of any one number to score in this category, and, as above, you score is the sum of all your dice. Again, you may put a five-of-a-kind (a Yahtzee) in here, too.

- *Full house.* To win points in this category, you must roll two of one number and three of another. A full house is worth 25 points, regardless of which two numbers you have. (A five-of-a-kind is a full house, too, since you would have two of one number and three more of the same number.)

- *Small straight.* A small straight is any four dice in a row, either 1-2-3-4-5 or 2-3-4-5-6. It is worth 40 points. As with small straights, the dice do not need to be in order, so that 3-1-5-2-4 counts as a straight.

- *Yahtzee.* You get a Yahtzee (or a five-of-a-kind) when all five dice show the same number. The first time that you roll this, you get to score 50 points. If you can roll more Yahtzees, you get 100 more points for each one you roll.

- *Chance.* Chance is simply the sum of all five dice, regardless of what they are. This is useful when your turn does not produce any winning combinations, and you don't want to use the score anywhere else.

If your turn does not produce any useful combinations, you must put a zero in as one of your scores. There are two ways to look at this. One is that you should put a zero into a score that would not have given you very many points anyway, such as the one's sub-category, which is worth only 5 points maximum. However, another way to view this situation is that you should put a zero into a relatively unlikely (and consequently high-scoring) category, on the assumption that you might not be able to win points for that

¹ "Yahtzee" is a registered trademark of E. S. Lowe Company, Inc.

combination anyway. Which you choose depends on how you play.

This program allows up to four people to play at once. Each player enters his or her name, and play begins. To start your turn, press [RETURN] to roll your dice. From this initial combination of five dice, all possible scores will be computed automatically and displayed. When you want to enter a score, you press the letter corresponding to that category, and that score will be taken as the permanent score.

For example, if your first roll was 5-4-5-3-2, then there would be a 2 in the two's category, and 3 under "threes," a 4 for "fours," and a 19 in "chance." No other categories would get points. You must now decide what to do. You could press [J] to score the small straight, or reroll the 4-3-2 dice to try for more fives, or you could roll other dice to try for something else. To reroll a die, you simply enter the number showing on the

die that you no longer want. To reroll several dice, type together the numbers on all of them. If your turn does not produce a score, and you are forced to accept a zero, then you enter the letter of the category for which you want to enter a "zero." In any case, press [RETURN] to tell the PET that you've made your choice.

If you choose to zero a category, the PET will double-check by asking if you really want to zero it. Usually, you'll answer "yes." Occasionally, you will have accidentally pressed the wrong letter and this question helps protect against that. You can prevent the question by typing a "z" (for zero) before the category letter, as "ZL" to zero category L.

When you have completed your turn, the next player begins his. This continues until each player has had 13 rolls. The grand totals are then displayed for all to see, in order from highest to lowest.



```

0 CLR:POKE 59468,12
6 REM  YAHTZEE COPYRIGHT (C) 1981 THE CODE WORKS
10 REM  10/24/81
90 PG$="YAHTZEE":NM$="9":GOSUB 62000
100 CLR:DIM DI(5),DI$(6,6),T(5),TS(13),SC$(3,13),TL(4,3),NM$(4)
105 PRINT "{clr}"CHR$(142)
110 FOR I=1 TO 6:FOR J=1 TO 6:READ DI$(I,J):NEXT J,I
120 DATA "___","_{Q}","{Q}_","{Q}__{Q}","{Q}__{Q}","{QQQ}","_{Q}_",
"___","_{Q}_","___","_{Q}_",
121 DATA "___","___","{Q}_","_{Q}_","{Q}__{Q}","{Q}__{Q}","{QQQ}",
"___","{Q}_","_{Q}_","{Q}__{Q}"
122 DATA "{Q}__{Q}","{Q}__{Q}","_{Q}_","___","_{Q}_","___","_{Q}_",
"{Q}__{Q}","___","_{Q}_","{Q}_",
123 DATA "{Q}__{Q}","{Q}__{Q}","{Q}__{Q}"

```



```

130 DN$="{home 7°down}"
200 PRINT "HOW_MANY_PLAYERS?_";:GOSUB 60000:PRINT
210 NP=VAL(IN$):IF NP<1 OR NP>4 THEN
    PRINT "ONE_TO_FOUR_PLAYERS,_PLEASE.":GOTO 200
220 PRINT "{down}ENTER_THEIR_NAMES,_PLEASE.":
    PRINT "(FIVE_LETTERS_LONG,_MAX.){down}"
240 FOR I=1 TO NP:PRINT "PLAYER"I":_";:GOSUB 60000:
    NM$(I-1)=LEFT$(IN$,12):PRINT :NEXT
300 FOR I=0 TO 3:TL(I,0)=15+6*I+5-LEN(NM$(I))
400 FOR J=1 TO 13:SC%(I,J)=-1:NEXT J:FOR J=1 TO 3:TL(I,J)=0:
    NEXT J,I
900 PL=NP:RD=0:GOSUB 10000:PRINT DN$,"{up}";
920 FOR I=0 TO NP-1:PRINT TAB(TL(I,0));NM$(I);"{up}":NEXT I
1500 FOR I=1 TO 5:DI(I)=0:NEXT I:RL=0
1520 PL=PL+1:IF PL>=NP THEN PL=0:RD=RD+1:IF RD>13 THEN 5000
1530 GOSUB 11000
1540 PRINT DN$,"{up rvs}ROUND";RD;TAB(TL(PL,0));NM$(PL)
2000 PR$="?_":GOSUB 9100:IF RL=0 THEN IN$="00000"
2010 IF IN$="" THEN 2000
2020 ZER=0:IF LEFT$(IN$,1)="Z" THEN ZER=1:IN$=MID$(IN$,2):
    IF IN$="" THEN 2000
2030 N=1:A=1:FOR I=1 TO LEN(IN$)
2040 T$=MID$(IN$,I,1):IF T$<"A" OR T$>"M" THEN A=0
2050 IF T$<"0" OR T$>"6" THEN N=0
2060 NEXT I
2070 IF A THEN 3000
2080 IF N=0 THEN 2000
2090 RL=RL+1:IF RL>3 THEN PR$="NO_MORE_ROLLS_ALLOWED.":GOTO 3500
2100 FOR I=1 TO LEN(IN$):T=VAL(MID$(IN$,I,1))
2105 FOR J=1 TO 5:IF DI(J)=T THEN DI(J)=0:N=0:GOTO 2115
2110 NEXT J
2115 NEXT I:IF N THEN RL=RL-1:GOTO 3500
2117 PRINT "{home down rvs}ROLL{3°left down}";RL
2120 GOSUB 9000:GOSUB 8000:RV$="{rvs}":PRINT DN$;
2130 FOR CT=1 TO 6:GOSUB 2600:PRINT :NEXT CT
2131 PRINT :PRINT
2132 FOR CT=7 TO 13:GOSUB 2600:PRINT :NEXT CT
2133 GOTO 2000
2140 FOR I=1 TO 6:SC=TS(I):IF SC<>0 AND SC%(PL,I)<0 THEN GOSUB 2500:
    PRINT "{up}";
2150 PRINT :NEXT I:PRINT :PRINT
2160 FOR I=7 TO 13:SC=TS(I):IF SC<>0 AND SC%(PL,I)<0 THEN
    GOSUB 2500:PRINT "{up}";
2170 PRINT :NEXT I:GOTO 2000
2500 PRINT TAB(17+6*PL);RV$;RIGHT$("___"+STR$(SC),3):RETURN
2510 PRINT TAB(17+6*PL);"__":RETURN
2600 IF SC%(PL,CT)>=0 THEN RETURN
2610 SC=TS(CT)
2620 ON (SC=0)+2 GOSUB 2510,2500
2630 PRINT "{up}";:RETURN
3000 RV$="":T=ASC(IN$)-64:ADD=TS(T)

```

```

3005 IF T=12 AND ADD>0 AND SC%(PL,T)>0 THEN ADD=100:GOTO 3100
3010 IF SC%(PL,T)>=0 THEN PR$="YOU'VE ALREADY TAKEN IT.":GOTO 3500
3020 IF ADD>0 OR ZER=1 THEN 3100
3030 PR$="REALLY ZERO IT?":GOSUB 9100
3040 IF LEFT$(IN$,1)<>"Y" THEN 2000
3100 X=T:IF T>6 THEN X=X+2
3105 IF SC%(PL,T)<0 THEN SC%(PL,T)=0
3110 PRINT :PRINT DN$;LEFT$("{off 14°down}",X);
3120 RV$="":SC=ADD+SC%(PL,T):SC%(PL,T)=SC
3130 GOSUB 2500:IF T>6 THEN 3300
3210 PRINT LEFT$("{6°down}",7-T);:X=TL(PL,1):SC=X+ADD:TL(PL,1)=SC
3220 IF X>62 OR SC<63 THEN 3250
3230 X=SC:SC=35:X=X+SC:PRINT "{up}";:GOSUB 2500:TL(PL,1)=X
3240 TL(PL,3)=TL(PL,3)+SC:SC=X
3250 GOSUB 2500:PRINT "{8°down}";:GOTO 3400
3300 PRINT LEFT$("{off 7°down}",14-T);
3310 SC=TL(PL,2)+ADD:TL(PL,2)=SC:GOSUB 2500
3400 SC=TL(PL,3)+ADD:TL(PL,3)=SC:RV$="":GOSUB 2500
3450 PRINT DN$;:FOR I=1 TO 6:IF SC%(PL,I)<0 THEN GOSUB 2510:
PRINT "{up}";
3455 PRINT :NEXT I
3460 PRINT "{down}":FOR I=7 TO 13:IF SC%(PL,I)<0 THEN GOSUB 2510:
PRINT "{up}";
3465 PRINT :NEXT I
3470 PRINT DN$;"{up}";TAB(TL(PL,0));NM$(PL):GOTO 1500
3500 PRINT DN$;"{2°up}";PR$:FOR I=1 TO 1000:NEXT I:GOTO 2000
5000 TM=TI+300:FOR I=0 TO NP-1:FOR J=0 TO NP-2
5010 IF TL(J,3)>TL(J+1,3) THEN 5040
5020 T=TL(J,3):TL(J,3)=TL(J+1,3):TL(J+1,3)=T
5030 T$=NM$(J):NM$(J)=NM$(J+1):NM$(J+1)=T$
5040 NEXT J,I
5050 IF TI<TM THEN 5050
5060 PRINT "{clr down}FINAL STANDINGS:{down}"
5070 FOR I=0 TO NP-1:PRINT TL(I,3);TAB(5);NM$(I):NEXT I
5100 PRINT "{down}PLAY AGAIN?":GOSUB 60000
5110 IF LEFT$(IN$,1)<>"N" THEN 300
5120 END
8000 FOR I=1 TO 13:TS(I)=0:NEXT I:CT=0:SUM=0
8010 FOR I=1 TO 5:T=DI(I):X=TS(T):IF X=0 THEN CT=CT+1
8020 TS(T)=X+1:SUM=SUM+T:NEXT I:TS(13)=SUM
8050 ON CT GOTO 8100,8200,8150,8300,8400
8100 TS(12)=50:TS(8)=SUM:TS(7)=SUM:GOTO 8500
8150 FOR I=1 TO 6:IF TS(I)=3 THEN TS(7)=SUM
8160 NEXT I:GOTO 8500
8200 CT=0:FOR I=1 TO 6:IF TS(I)=1 THEN CT=CT+1
8220 NEXT I:TS(7)=SUM
8240 IF CT=0 THEN TS(9)=25
8250 IF CT=1 THEN TS(8)=SUM
8260 GOTO 8500
8300 MN=7:MX=0:FOR I=1 TO 6
8310 IF TS(I)=0 THEN 8340

```

```

8320 IF I<MN THEN MN=I
8330 IF I>MX THEN MX=I
8340 NEXT I
8350 IF MX-MN=3 THEN TS(10)=30
8360 GOTO 8500
8400 IF TS(1)=0 OR TS(6)=0 THEN TS(11)=40:TS(10)=30:GOTO 8500
8410 IF TS(2)=0 OR TS(5)=0 THEN TS(10)=30:GOTO 8500
8420 GOTO 8500
8430 IF I>MX THEN MX=I
8440 NEXT I
8450 IF MX-MN=4 THEN TS(10)=30:TS(11)=40
8500 FOR I=1 TO 6:TS(I)=TS(I)*I:NEXT I
8510 RETURN
9000 PRINT "{home}";
9010 FOR I=1 TO 3:PRINT :FOR J=1 TO 5:IF DI(J)=0 THEN
    PRINT TAB(7*J)"...";
9020 NEXT J,I:FOR I=1 TO 5
9030 IF DI(I)=0 THEN Q=INT(RND(1)*6+1):T(I)=3*(RND(1)>.5)+3:
    CT(Q)=CT(Q)+1:DI(I)=Q
9040 NEXT I
9050 PRINT "{home}";:FOR I=1 TO 3:PRINT
9060 FOR J=1 TO 5:PRINT TAB(7*J);DI$(I+T(J),DI(J));:NEXT J,I
9070 PRINT :RETURN
9100 PRINT DN$"{2°up}.....{up}":
    PRINT PR$;
9110 GOSUB 60000:RETURN
10000 PRINT "{clr}":GOSUB 11000
10020 T$=="TOTAL"=====
10030 PRINT DN$;"A:ACES":PRINT "B:TWOS":PRINT "C:THREES"
10040 PRINT "D:FOURS":PRINT "E:FIVES":PRINT "F:SIXES"
10050 PRINT "...>=63_BONUS":PRINT T$
10060 PRINT "G:3_OF_A_KIND":PRINT "H:4_OF_A_KIND":
    PRINT "I:FULL_HOUSE"
10070 PRINT "J:SMALL_STRAIGHT":PRINT "K:LARGE_STRAIGHT"
10080 PRINT "L:YAHTZEE":PRINT "M:CHANCE"
10090 PRINT T$:PRINT "...GRAND_TOTAL"
10100 RETURN
11000 T$="...{U@@@I}":U$="{2°right }...{I}":V$="...{J@@@K}":
    S$="{4°right}"
11010 PRINT "{home}"S$T$T$T$T$T$:FOR I=1 TO 3
11020 PRINT S$U$U$U$U$U$:NEXT I:PRINT S$V$V$V$V$V$;"{3°up}"
11040 PRINT "{home down rvs}.....{4°left down}.....{4°left down}....."
11050 RETURN
60000 IN$="...":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "...{left}";:
    GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
    Z$=CHR$((Z+128) AND 255)

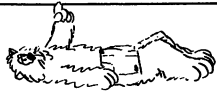
```

```

60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60205 GOSUB 60500
60210 PRINT "PRESS_{rvs}RETURN{off}_{TO}";
60220 IF PN=LP THEN PRINT "BEGIN:";:GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_{-}_CURSOR_{#}";NM$;
"_{_}(C)_{_}1981_{_}_{_}PAGE";PN:GOSUB 60500
60270 RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;"_{_}BY_{_}GLEN_{_}FISHER"
62002 PRINT "{down}_{_}_{_}_{_}_{_}CURSOR_{_}#9_{_}_{_}COPYRIGHT_{_}(C)_{_}1981"
62005 GOSUB 60500
62010 PRINT "{down}THE_{_}GAME_{_}OF_{_}YAHTZEE"
62020 PRINT "{2°down}WANT_{_}DIRECTIONS?_{_}";:GOSUB 60000
62030 IF LEFT$(IN$,1)<>"Y" THEN 100
62100 LP=2:GOSUB 60200
62110 PRINT "DICE_{_}COMBINATIONS:"
62120 PRINT "{down}ACES_{_}THRU_{_}SIXES:_{_}ONE{_{_}}OR_{_}MORE_{_}OF_{_}THE"
62130 PRINT "APPROPRIATE_{_}NUMBER_{_}(E.G._{_}ONE_{_}4_{_},_{_}OR"
62140 PRINT "THREE_{_}2'S)_{_}_{_}WORTH_{_}THE_{_}SUM_{_}OF_{_}THE_{_}DICE"
62150 PRINT "OF_{_}THAT_{_}NUMBER_{_}(3_{_}3_{_}4_{_}5_{_}3_{_}IS_{_}WORTH_{_}9)_{_}."
62160 PRINT "{down}THREE_{_}OF_{_}A_{_}KIND:_{_}ANY_{_}THREE_{_}DICE_{_}OF_{_}THE"
62170 PRINT "SAME_{_}NUMBER_{_}_{_}WORTH_{_}THE_{_}SUM_{_}OF_{_}ALL_{_}FIVE"
62180 PRINT "DICE_{_}."
62190 PRINT "{down}FOUR_{_}OF_{_}A_{_}KIND:_{_}ANY_{_}FOUR_{_}DICE_{_}OF_{_}THE"
62200 PRINT "SAME_{_}NUMBER_{_}_{_}ALSO_{_}WORTH_{_}THE_{_}SUM_{_}OF_{_}ALL"
62210 PRINT "FIVE_{_}DICE_{_}."
62220 PRINT "{down}FULL_{_}HOUSE:_{_}THREE_{_}OF_{_}ONE_{_}NUMBER_{_},_{_}TWO"
62230 PRINT "OF_{_}ANOTHER_{_}_{_}WORTH{_{_}}25_{_}POINTS_{_}."
62240 GOSUB 60200
62250 PRINT "SMALL_{_}STRAIGHT:_{_}A_{_}RUN_{_}OF_{_}FOUR_{_}DICE_{_},_{_}AS"
62260 PRINT "2_{_}3_{_}4_{_}5_{_}OR_{_}2_{_}4_{_}1_{_}3_{_}_{_}WORTH{_{_}}30_{_}POINTS_{_}."
62270 PRINT "{down}LARGE_{_}STRAIGHT:_{_}A_{_}RUN_{_}OF_{_}ALL_{_}FIVE_{_}DICE_{_}."
62280 PRINT "WORTH{_{_}}40_{_}POINTS_{_}."
62290 PRINT "{down}YAHTZEE:_{_}ALL_{_}FIVE_{_}DICE_{_}OF_{_}THE_{_}SAME"
62300 PRINT "NUMBER_{_}_{_}WORTH{_{_}}50_{_}FOR_{_}THE_{_}FIRST_{_}ONE_{_},_{_}AND"
62310 PRINT "100_{_}FOR_{_}EACH_{_}ONE_{_}THEREAFTER_{_}."
62320 PRINT "{down}CHANCE:_{_}ANYTHING_{_}AT_{_}ALL_{_}_{_}WORTH_{_}THE_{_}SUM"
62330 PRINT "OF_{_}THE_{_}DICE_{_}."
62340 PRINT "{down}TO_{_}ROLL_{_}THE_{_}DICE_{_},_{_}TYPE_{_}THE_{_}NUMBERS_{_}ON"
62350 PRINT "THE_{_}DICE_{_}TO_{_}ROLL_{_}_{_}TO_{_}SCORE_{_},_{_}TYPE_{_}THE"

```

```
62360 PRINT "LETTER OF THE COMBINATION."  
62365 PRINT "TO ZERO A COMBINATION, TYPE A{ } 'Z' "  
62366 PRINT "FOLLOWED BY THE LETTER."  
62370 GOSUB 60200
```



BONZO

By Randall Lockwood

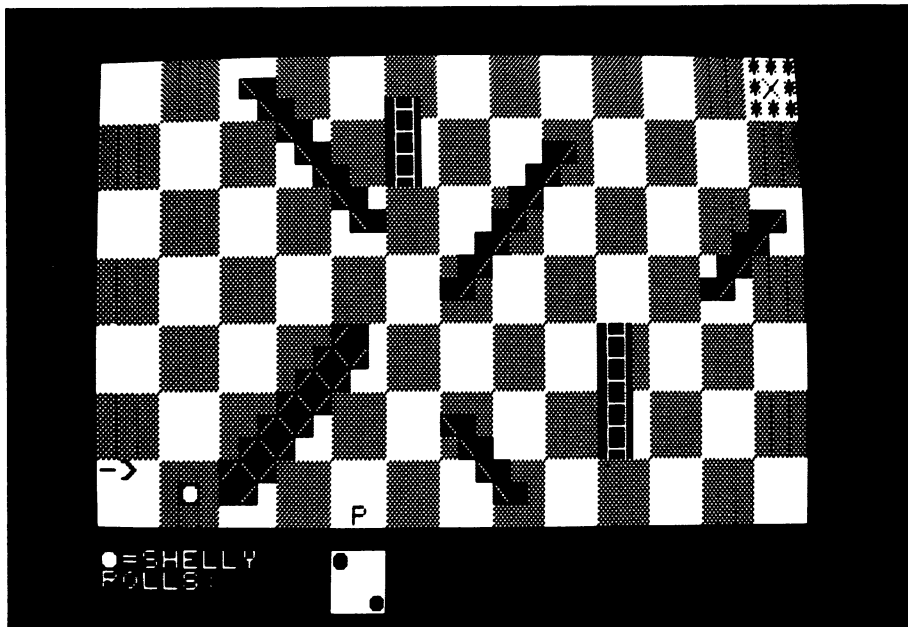


Remember the old game, "Bonzo Dog Chase"? (Well, perhaps you know it as "Chutes and Ladders"? BONZO brings that enjoyable game to the screen of your PET. The board is an expanse of black-and-white squares, with slides and ladders running between several of the squares. The game itself is a race to the top, with ladders as shortcuts and slides as hazards. Land on a square at the base of a ladder, and you climb the ladder to the top, skipping all the

squares in between. But, land on a square at the top of a chute, and you slide all the way down to the bottom, and must travel all that distance again.

You can play against the PET or against a friend, or against both. When the PET tells you that it is your turn, the die will roll until you press [RETURN]. The number that turns up is the number of squares that you may move and the PET will move you automatically.

¹ "Chutes and Ladders" is a registered trademark of Milton Bradley Co.



```
0 CLR:POKE 59468,12
6 REM BONZO! COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81 RJ
90 PG$="BONZO!":NM$="16":GOSUB 62000
100 DIM O(9),L(9)
110 PRINT "THERE WILL BE A SHORT DELAY.":
PRINT "{down}PLEASE BE PATIENT."
120 I=RND(-TI)
140 POKE 59467,16:POKE 59466,15:SN=59464:POKE SN,0
150 DIM A(3),P(91),Z%(91),M(3),S(3)
160 M(1)=81:M(2)=218:M(3)=144
165 X(1)=0:X(2)=WD:X(3)=WD+1:W(1)=50:W(2)=80:W(3)=120
170 FOR I=0 TO 12
180 FOR J=0 TO 6 STEP 2:P(J*13+I+1)=CRT+WD*19+1-WD*3*J+3*I:NEXT J
```

```

190 FOR J=1 TO 5 STEP 2:P(J*13+I+1)=CRT+WD*19+37-WD*3*J-3*I:NEXT J
200 NEXT I
210 VT=22
220 BL$="{up}.....{up}"
230 READ T$:IF T$<>"{" THEN 230
240 FOR I=826 TO 848:READ N:POKE I,N:NEXT
250 FOR I=1 TO 9:READ L(I):NEXT I:O(1)=-WD:O(2)=-WD:O(3)=-WD+1:
  O(4)=-WD+1:O(5)=-WD-1
255 O(6)=WD+1:O(7)=WD+1:O(8)=WD-1:O(9)=WD-1
260 FOR I=0 TO 9:READ D$(I):NEXT I
300 PRINT "{clr 4°down}HOW_MANY_{rvs}PEOPLE{off}_ARE_PLAYING?":
  PRINT "{down}(1_OR_2)?_";
310 GOSUB 60000:NP=VAL(IN$):IF NP<1 OR NP>2 THEN 300
320 FOR I=1 TO NP
330 PRINT "{down}WHAT_IS_PLAYER_{I};I;'S_NAME?_";:GOSUB 60000:
  N$(I)=IN$
335 IF IN$="" THEN PRINT "{down}NO_ANONYMOUS_PLAYERS_ALLOWED!":
  GOTO 330
337 IF LEN(IN$)>10 THEN
  PRINT "{down}NAMES_CAN'T_BE_LONGER_THAN_10_LETTERS!":GOTO 330
340 NEXT I:N$(3)="PET"
350 PRINT "{down}CAN_I_PLAY,_TOO?_";:GOSUB 60000:IF LEFT$(IN$,
  1)="N" THEN 410
355 IF IN$="" THEN PRINT TAB(17);"{up}YES":FOR I=1 TO 500:NEXT
390 NP=NP+1:N$(NP)="PET":M(NP)=144
410 B1$="{rvs}...{down 3°left}...{down 3°left}...{off 2°up}"
420 B2$="{&&& down 3°left &&& down 3°left &&& 2°up}"
430 L$(1)="{+3 down 2°left +3 down 2°left +3 down 2°left +3}"
440 L$(2)="{+3 down 2°left +3 down 2°left +3 down 2°left +3 down
  2°left +3 down 2°left +3}"
450 L$(3)="_/{up 2°left}/{}/{up 2°left}/{}/{up 2°left}/{}/{up
  2°left}/{_}"
460 L$(4)="_/{up 2°left}/{}/{up 2°left}/{}/{up 2°left}/{}/{up
  2°left}/{}/{up 2°left}/{}/{up 2°left}/{}/{up 2°left}/{_}"
470 L$(5)="__{down 2°left}/{}/{down 2°left}/{}/{down 2°left}_"
480 L$(6)="__{down left} __{down left} __{down left} __"
490 L$(7)="__{down left} __{down left} __{down left} __{down
  left} __{down left} __{down left} __"
500 L$(8)="_/{down 3°left} _/{down 3°left} _/{down 3°left} _/"
510 L$(9)="_/{down 3°left} _/{down 3°left} _/{down 3°left} _/{down
  3°left} _/{down 3°left} _/{down 3°left} _/"
515 FOR J=1 TO 10:E(J)=0:NEXT J:FOR J=1 TO 91:Z%(J)=0:NEXT J:
  PRINT "{clr}";
520 FOR J=1 TO 3:FOR I=1 TO 6:PRINT B1$;B2$;:NEXT:
  PRINT B1$"{2°down}"
530 FOR I=1 TO 6:PRINT B2$;B1$;:NEXT:PRINT B2$"{2°down}":NEXT J
540 FOR I=1 TO 6:PRINT B1$;B2$;:NEXT:PRINT B1$"{2°down}"
550 POKE QL,19:PRINT "{up rvs}->":PRINT "{home}";TAB(36);
  "{rvs}***{down 3°left}*{V}*{down 3°left}***"
560 RESTORE:Z=INT(RND(1)*4)+1
570 IF Z>1 THEN FOR K=1 TO Z-1:READ N:FOR J=1 TO N:READ T$:NEXT J,K

```

```

580 READ N:FOR K=1 TO N:READ I$:GOSUB 1080:NEXT K
590 DATA 7,162143462,030617681,032468769
600 DATA 073324167,183151216,061397450,120375123
610 :
620 DATA 7,130622450,152163418,223041139,131234858
630 DATA 030798153,032178662,033489068
640 :
650 DATA 9,181050523,152411835,153133741,121884632
660 DATA 160642957,040125379,032198658,032568769,033078965
670 :
680 DATA 7,210640331,181862008,142721743
690 DATA 093686541,062497046,030778157,041517384
700 :
710 FOR K=1 TO NP:S(K)=1:A(K)=160:POKE P(1)+X(K),M(K):NEXT K
720 FOR K=1 TO NP
730 SK=S(K):MK=M(K):XK=X(K):PSK=P(SK)
740 GOSUB 770
750 GOSUB 910
760 NEXT K:GOTO 720:END
770 D=INT(RND(1)*6)+1:POKE QL,VT+1
780 PRINT "{up}";TAB(12);D$(0);"{2°up}":PRINT BL$
785 POKE CRT+WD*22,M(K):PRINT "{right}=";N$(K):PRINT "ROLLS:"
790 IF N$(K)="PET" THEN 830
795 GET T$:IF T$<>" THEN 795
800 GOSUB 880
810 GET T$:IF T$="" THEN 800
815 IF T$="Q" THEN 1070
820 GOTO 860
830 FOR J=1 TO INT(RND(1)*10)+3
840 GOSUB 880
850 NEXT J
860 Z=D:GOSUB 890:FOR I=1 TO 1000:NEXT I
870 RETURN
880 Z=INT(RND(1)*10)+1
890 POKE QL,VT+1:PRINT TAB(12);"{up rvs}";D$(Z);"{up}"
900 FOR I=1 TO 100:NEXT I:RETURN
910 POKE PSK+XK,A(K):FOR I=1 TO D
920 IF SK+I>=91 THEN 1000
930 A(K)=PEEK(P(SK+I)+XK)
940 POKE P(SK+I)+XK,M(K):POKE SN,W(K):FOR J=1 TO 50:NEXT J:
    POKE SN,0
950 IF I<D THEN POKE P(SK+I)+XK,A(K)
960 NEXT I
970 POKE SN,0:SK=S(K)+D:S(K)=SK:PSK=P(SK)
980 U=Z%(SK):IF U=0 THEN RETURN
990 GOTO 1270
1000 POKE QL,VT:PRINT BL$:PRINT TAB(5);N$(K)"_WINS!!!"
1010 FOR I=1 TO 99:R%=100*RND(1):POKE SN,30+R%:NEXT:POKE SN,0
1020 IF WD=40 THEN FOR I=1 TO 8:SYS 826:FOR J=1 TO 300:NEXT:NEXT
1030 FOR I=1 TO 1000:NEXT
1040 POKE QL,VT:PRINT BL$:PRINT "PLAY_AGAIN?";

```



```
1050 GOSUB 60000
1060 IF LEFT$(IN$,1)<>"N" THEN 515
1070 POKE 59467,0:END
1080 Y=VAL(LEFT$(I$,2)):X=VAL(MID$(I$,3,2))
1090 Z=VAL(MID$(I$,5,1)):S=VAL(MID$(I$,6,2)):E(Z)=VAL(RIGHT$(I$,2))
1100 POKE QL,Y-1:PRINT TAB(X);"{up}";L$(Z):Z%(S)=Z:RETURN
1270 O=O(U):L=L(U)
1280 LP=100:HP=200:Z=1:IF U<=5 THEN LP=200:HP=100:Z=-1
1290 IV=INT((HP-LP)/L+.5)
1300 S1=PSK+X(K):S2=A(K):FOR I=1 TO L
1310 POKE S1,S2:S1=S1+O:S2=PEEK(S1):POKE S1,M(K)
1320 FOR J=LP TO LP+IV STEP Z:POKE SN,J:NEXT:LP=LP+IV
1330 NEXT I
1340 POKE SN,0
1350 SK=E(U):S(K)=SK:A(K)=S2
1360 RETURN
```



BOSWAIN

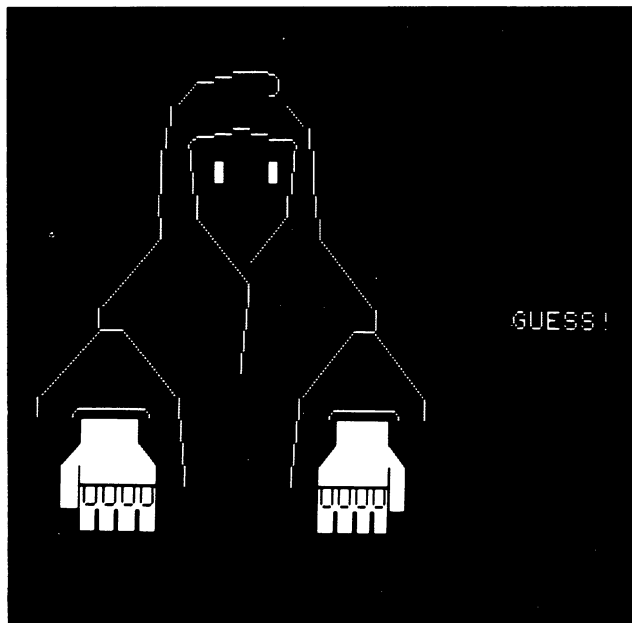
By Chris Nadovich

TANSTAAFL — There Ain't No Such Thing As A Free Lunch. In other words, everything must be paid for, in its own way. Games of chance are no different, and the time has come to pay!

BOSWAIN is the guardian of the Temple of

Yakban, and you are now his captive. BOSWAIN is willing to set you free again — for a price. You may pay in rubies (if you can win them from him), or you may pay with something else.

C.T. Nadovich truly deserves a hand for this program.



```
0 CLR:POKE 59468,12
6 REM BOSWAIN COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81 RJ
90 PG$="BOSWAIN":NM$="21":GOSUB 62000
100 L1=0:DY=0:DX=1:SC=0:LH$="*":RH$="_"
101 PRINT CHR$(142)
170 DL=2:GOSUB 6000:NH=1:GOSUB 11200:GOSUB 1000:DL=.5:GOSUB 6000
200 DL=.5:GOSUB 6000:HS=0:GOSUB 2000
210 IF PP=0 THEN RH$="*":LH$="_"
215 IF PP=1 THEN RH$="_" :LH$="*"
220 GOSUB 4000:POKE QL,12:PRINT "{up}":PRINT TAB(29)"GUESS!":
    GOSUB 6500
250 IF Q$<>"4" AND Q$<>"6" THEN GOSUB 1700:GOTO 200
270 LP=0:IF Q$="6" THEN LP=1
280 IF LP=PP THEN SC=SC-1
290 IF LP<>PP THEN SC=SC+1
300 HS=1:IF Q$="4" THEN HS=2
320 GOSUB 2000:GOSUB 7000
```

```

330 IF HS<>5 THEN HS=3:GOSUB 2000:DL=1:GOSUB 6000:HS=5:GOSUB 2000:
    GOSUB 4000
340 IF SC=10 GOTO 600
350 IF SC=-10 GOTO 400
360 GOSUB 11200:GOTO 200
400 RESTORE
410 READ Q$:IF Q$<>"*" THEN 410
430 GOSUB 5000:DL=2:GOSUB 6000
440 GOSUB 4900:GOSUB 13000
450 DL=5:GOSUB 6000:PRINT "{clr 2°down}":END
600 RH$="._":LH$="._":HS=3:GOSUB 2000
620 Q$="YOU HAVE WON!":Y=14:GOSUB 5100
630 Q$="YOU ARE FREE!":Y=12:GOSUB 5100
640 PRINT "{home}":END
1000 PRINT "{clr down}.....OH MY!!..YOU SEEM TO HAVE FALLEN"
1060 PRINT "{down}..THROUGH A TRAP DOOR....WHEN YOUR HEAD"
1070 PRINT "{down}..STOPS SPINNING,{.}YOUR NEW SURROUNDINGS"
1075 PRINT "{down}..COME INTO FOCUS."
1080 PRINT "{down}.....ABOVE YOU IS A VAST DOMED IVORY"
1090 PRINT "{down}..CEILING, TRACED WITH CRIMSON STREAKS."
1100 PRINT "{down}..AT ITS PEAK IS A ROUND BLACK OPENING,"
1110 PRINT "{down}..PRESUMABLY..THROUGH..WHICH..YOU FELL."
1120 PRINT "{down}..YOUR FEET TELL YOU..THAT..THE{.}..FLOOR,"
1130 PRINT "{down}..HIDDEN..BY..{.}WAIST HIGH MIST,..{.}IS SOFT"
1140 PRINT "{down}..AND RESILIENT."
1160 GOSUB 6400
1170 PRINT "{clr 7°down}.....AS YOU BEGIN TO..MAKE{.}..YOUR..WAY"
1180 PRINT "{down}..TOWARD THE ARCHED WALLS, A NOISE FROM"
1190 PRINT "{down}..BEHIND STARTLES YOU..YOU SPIN TO FACE"
1200 PRINT "{down}..THE INHABITANT OF THIS ERIE PLACE,..A"
1210 PRINT "{down}..MYSTERIOUS HOODED FIGURE."
1230 GOSUB 6400
1240 PRINT "{clr}":GOSUB 9000:HS=5:GOSUB 2000
1241 RESTORE:FOR I=1 TO 3
1243 GOSUB 4900:IF I<3 THEN DL=2:GOSUB 6000
1245 NEXT I:GOSUB 6500:IF Q$="N" THEN 1600
1255 GOSUB 4900:DL=2:GOSUB 6000:GOSUB 4000:HS=3:GOSUB 2000:
    GOSUB 5000
1290 DL=.5:GOSUB 6000:HS=0:GOSUB 2000:DL=1.5:GOSUB 6000:GOSUB 4900
1310 DL=2:GOSUB 6000:GOSUB 1700:Y=16:GOTO 1610
1600 GOSUB 4000:Y=16:Q$="GOOD!":GOSUB 5100:Y=Y-3
1610 Q$="LET US":GOSUB 5100:Y=Y-2:Q$="BEGIN!":GOSUB 5100:DL=3:
    GOSUB 6000
1640 GOSUB 4000:RETURN
1700 PRINT "{clr
    7°down}.....{rvs}NOTE FROM OMNISCIENT NARRATOR{off}"
1710 PRINT "{2°down}.....YOU MAY INDICATE BOSWAIN'S HANDS"
1720 PRINT "{down}..BY..USING THE 4 AND 6 KEYS..THE 4 KEY"
1730 PRINT "{down}..INDICATES HIS RIGHT HAND, 6 HIS LEFT."
1731 PRINT ".....{###}.....{###}"
1740 GOSUB 6400

```

```

1750 GOSUB 9000:HS=5:GOSUB 2000
1760 RETURN
2000 IF HS=5 THEN GOSUB 8000:CF=0:RETURN
2010 IF CF=0 THEN GOSUB 8500:CF=1
2020 ON HS GOTO 2150,2200,2250
2100 GOSUB 3200:GOSUB 3250:RETURN
2150 GOSUB 3200:GOSUB 3150:RETURN
2200 GOSUB 3100:GOSUB 3250:RETURN
2250 GOSUB 3100:GOSUB 3150:RETURN
3100 POKE QL,17:PRINT "{up}":PRINT "{2°right}";
3101 PRINT "...{rvs}...{off}...{down 7°left}";
3102 PRINT "...{rvs }...{off}...{down 7°left}";
3103 PRINT "...{rvs }..."LH$"...{off}...{down 7°left}";
3104 PRINT "{rvs ) off ) rvs 4}...{off}...{down 7°left}";
3105 PRINT "...{rvs 4444 off}...{down 7°left}";
3106 PRINT "...{rvs 4444 off}...{down 7°left}";
3107 PRINT "...{rvs 4444 off}...{down 7°left}";
3110 RETURN
3150 POKE QL,17:PRINT "{up}":PRINT TAB(18);
3151 PRINT "...{rvs}...{off}...{down 7°left}";
3152 PRINT "...{rvs}...{< off}...{down 7°left}";
3153 PRINT "...{rvs}..."RH$"...{< off}...{down 7°left}";
3154 PRINT "...{rvs}...{* off < rvs < off down 7°left}";
3155 PRINT "...{rvs **** off}...{down 7°left}";
3156 PRINT "...{rvs **** off}...{down 7°left}";
3157 PRINT "...{rvs **** off}...{7°left}";
3160 RETURN
3200 POKE QL,23:PRINT "{up}":PRINT "{4°right}";
3201 PRINT ".....{7°left up}";
3202 PRINT ".....{9°left up}";
3203 PRINT "...{rvs **'% off}...{up 8°left}";
3204 PRINT "...{rvs '}UUUU{off}...{up 8°left}";
3205 PRINT "...{rvs '$$$$ off}...{up 8°left}";
3206 PRINT "...{rvs )}...{< off}...{up 6°left}";
3207 PRINT "...{rvs}...{off}"
3210 RETURN
3250 POKE QL,23:PRINT "{up}":PRINT TAB(18);
3251 PRINT ".....{7°left up}";
3252 PRINT ".....{7°left up}";
3253 PRINT "{rvs '%44 off}...{up 8°left}";
3254 PRINT "{rvs }UUUU{% off}...{up 8°left}";
3255 PRINT "{rvs $$$$% off}...{up 8°left}";
3256 PRINT "{rvs )}...{< off}...{up 6°left}";
3257 PRINT "...{rvs}...{off}"
3260 RETURN
4000 PRINT "{home}";:FOR LJ=1 TO 24:PRINT TAB(25)".....":
NEXT:RETURN
4900 GOSUB 4000
5000 Y=16
5020 READ Q$:IF Q$<>"." THEN GOSUB 5100:Y=Y-2:GOTO 5020
5040 RETURN

```

```

5100 X=32-INT(LEN(Q$)/2)
5200 LN=LEN(Q$):LX=X:LY=Y
5210 FOR L=1 TO LN:POKE QL,25-Y:PRINT "{up}";
5250 PRINT TAB(LX)"*{left rvs}*{off left}";MID$(Q$,L,1)
5270 LX=LX+1:NEXT L
5290 RETURN
5510 DATA "I.AM","BOSWAIN!",.
5520 DATA "YOU.ARE.NOW","A.CAPTIVE","OF.THE.TEMPLE","YAKBAN",.
5540 DATA "NEED.I","ENLIGHTEN"
5550 DATA "YOU.ABOUT.THE","TERMS.OF.YOUR","RELEASE?",".
5570 DATA "TO.DEPART","YOU.MUST"
5580 DATA "PAY.A.TRIBUTE","OF.TEN.RUBIES",.
5590 DATA "IN.ONE.HAND","I.WILL.HIDE","A.GEM",.
5600 DATA "YOU.MUST","GUESS.WHICH","HAND.HOLDS","IT",.
5610 DATA =,"HA!!","YOU.HAVE","GUESSED.WRONG",.
5620 DATA "SINCE.YOU.HAVE","NO.JEWELS",.
5630 DATA "YOU.MUST.PAY","IN.FLESH!",.
5640 DATA "A.FINGER","IS.FAIR","EXCHANGE",.
5700 DATA *,"WILL.YOU","HONOR","YOUR.DEBT",,.
5710 DATA "OR.MUST","I.USE","FORCE?",".
6000 X=TI+60*DL
6020 IF TI<X GOTO 6020
6030 RETURN
6400 POKE QL,24:PRINT "{up}";TAB(25);"{rvs}PRESS.KEY..."
6500 GET Q$:IF Q$<>"GOTO 6500
6510 GET Q$:IF Q$="GOTO 6510
6520 RETURN
7000 GOSUB 4000
7100 IF L1=1 OR SC<>-1 GOTO 7300
7110 L1=1:RESTORE
7120 READ Q$:IF Q$<>"=" THEN 7120
7125 FOR I=1 TO 4
7130 GOSUB 5000
7135 IF I=1 THEN HS=3:GOSUB 2000
7140 DL=2:GOSUB 6000
7145 GOSUB 4000
7150 NEXT
7160 PRINT "{clr
      4°down}.....{rvs}NOTE.FROM.OMNISCIENT.NARRATOR{off}"
7170 PRINT "{2°down}.....AFTER.SEVERAL.MINUTES.OF.VIGOROUS"
7175 PRINT "{down}.ARGUMENT,.BOSWAIN.AGREEES.TO.ACCEPT"
7180 PRINT "{down}.YOUR.HANDS..AS..COLLATERAL..FOR..YOUR"
7185 PRINT "{down}.FINGERS..YOU.ARE.SAFE.FOR.NOW,.BUT..IF"
7190 PRINT "{down}.YOUR..DEBT.SHOULD.BECOME.TEN..FINGERS"
7195 PRINT "{down}.THEN.OFF.COME.YOUR.HANDS."
7196 PRINT ".....{#####}"
7205 GOSUB 6400
7210 GOSUB 9000:HS=5:GOSUB 2000:RETURN
7300 IF LP=PP GOTO 7308
7306 Q$="CORRECT!":Y=16:GOSUB 5100
7307 GOTO 7310

```

```

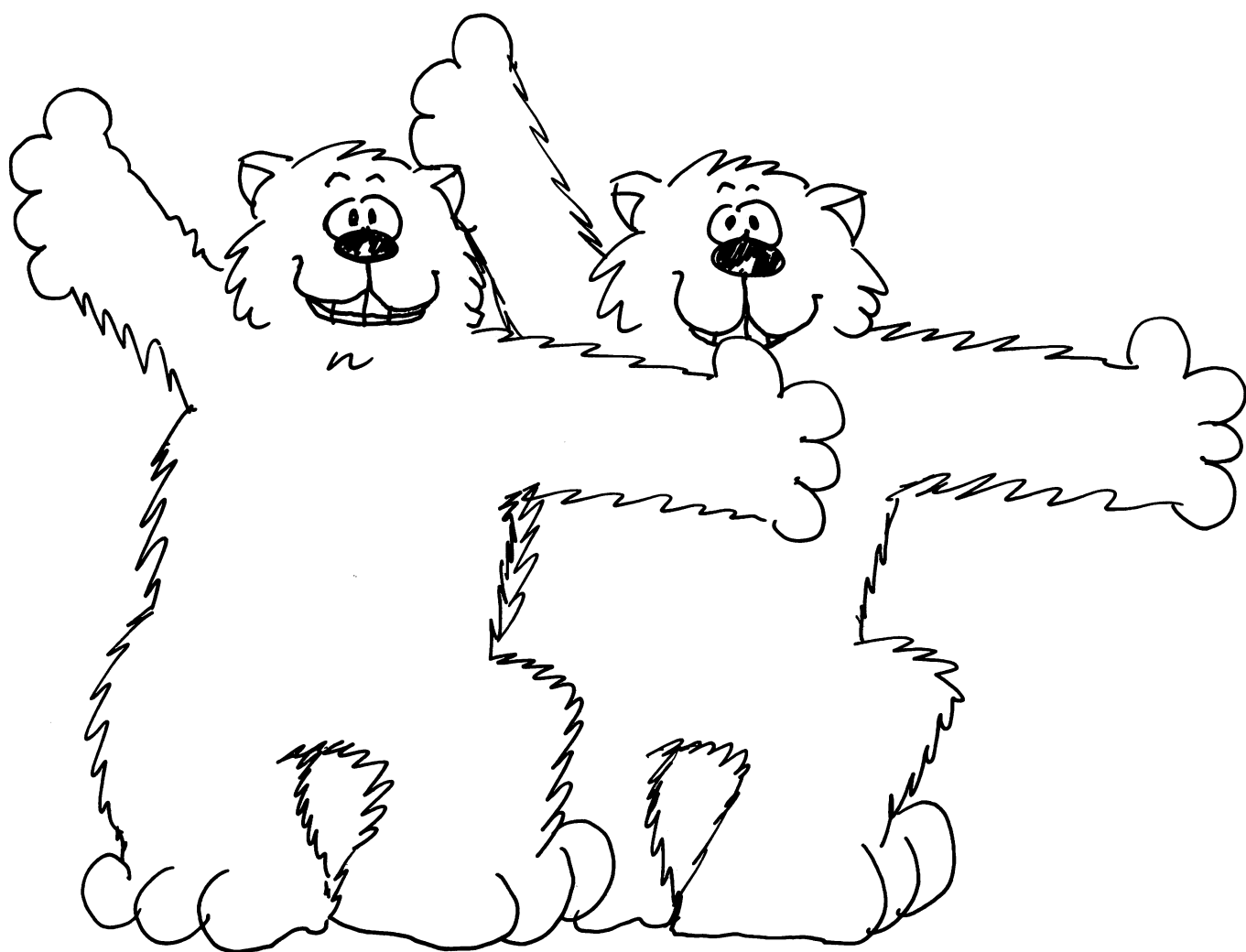
7308 Q$="INCORRECT!!":Y=16:GOSUB 5100
7310 IF SC>0 GOTO 7400
7311 IF SC=0 GOTO 7350
7315 Q$="YOU NOW OWE":Y=14:GOSUB 5100
7316 Q$=STR$(ABS(SC))+".FINGER"
7320 IF SC<>-1 THEN Q$=Q$+"S"
7325 Y=12:GOSUB 5100
7330 RETURN
7350 Q$="WE ARE EVEN":Y=12:GOSUB 5100
7355 RETURN
7400 Q$="YOU NOW HAVE":Y=14:GOSUB 5100
7405 R$=".RUBIES":IF SC=1 THEN R$=".RUBY"
7410 Q$=STR$(SC)+R$:Y=12:GOSUB 5100
7420 RETURN
8000 POKE QL,12:PRINT "{up}":PRINT ".....{Y}.....{Y}.....{T}"
8010 PRINT ".....{H}.....{H}.....{G}.."
8020 PRINT ".....{I}.....{I}.....{B}.."
8030 PRINT ".....{B}.....{B}.....{I}.."
8040 PRINT ".....{G}.....{B}.....{Y}.."
8050 PRINT ".....{' }.....{B}.....{' }.."
8060 PRINT ".....{Y}.....{G}.."
8070 PRINT ".....{Y}.....{G}.."
8080 PRINT ".....{Y}.....{G}.."
8085 PRINT ".....{Y}.....{G}.."
8086 PRINT ".....{Y}.....{G}.."
8087 PRINT ".....{Y}.....{G}.."
8090 RETURN
8500 POKE QL,12:PRINT "{up}":PRINT ".....{Y}.....{Y}.....{T}"
8501 PRINT ".....{N#M}.....{H}.....{N#M}"
8502 PRINT ".....{N}.....{M}.....{N}.....{M}"
8503 PRINT ".....{N}.....{M}.....{N}.....{M}"
8504 PRINT ".....{Y}.....{U@@@I}.....{T}.....{Y}.....{U@@@I}.....{G}"
8505 PRINT ".....{G}.....{H}....."
8506 PRINT ".....{B}.....{I}....."
8507 PRINT ".....{I}.....{B}....."
8508 FOR LI=1 TO 4:PRINT ".....":NEXT
8599 RETURN
9000 PRINT "{clr}"
9003 PRINT TAB(11) "${FCCI}"
9004 PRINT TAB(10) "${N}.....{K}"
9005 PRINT TAB(9) "${I}.....{M}"
9006 PRINT TAB(9) "${GUCD#D@IG}"
9007 PRINT TAB(9) "${GH}.....{; }.....{B}"
9008 PRINT TAB(9) "${GY}.....{>}.....{>G}"
9009 PRINT TAB(8) "${' }.....{GH}"
9010 PRINT TAB(9) "${G}.....{M}.....{N}.....{'}"
9011 PRINT TAB(8) "${N}.....{M}.....{N}.....{M}"
9012 PRINT TAB(7) "${N}.....{M}.....{M}"
9013 PRINT TAB(6) "${N}.....{' }.....{M}"
9100 RETURN
11200 IF NH<>1 GOTO 11300

```

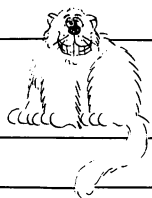
```

11210 FOR I=0 TO 7:TH(I)=0:NEXT I
11220 CC=1:NH=0
11300 I=-2*(.5+(LP=LL))
11320 J=4*OV(2)+2*OV(1)+1*OV(0)
11330 K=TH(J)+I
11335 IF K<>0 THEN K=K/ABS(K)
11340 IF CC>4 THEN TH(J)=K
11360 OV(2)=OV(1):OV(1)=OV(0)
11370 OV(0)=1+(PP=LP)
11390 J=4*OV(2)+2*OV(1)+1*OV(0)
11400 I=TH(J)
11410 IF I=0 GOTO 11450
11420 PP=(LP-.5)*I+.5
11430 GOTO 11500
11450 I=TI
11460 PP=2*((I/2)-INT(I/2))
11500 LL=LP:CC=CC+1:RETURN
13000 POKE QL,12:PRINT "{up}"
13002 PRINT TAB(19)"{N#M}"
13003 PRINT TAB(18)"{N}_{M}"
13004 PRINT TAB(17)"{N}.....{M}"
13005 PRINT TAB(16)"{Y}_{U@@@I}_{%}"
13006 PRINT TAB(16)"{H}_{rvs}....{off}"
13007 PRINT TAB(16)"{I rvs }....{< off}"
13008 PRINT TAB(16)"{B+ rvs 9}....{off I}"
13009 PRINT TAB(8)"{< rvs}_{FFF}....{6 off}_{rvs '%44 off l}"
13010 PRINT TAB(10)"{#7877##+@ rvs '%44 off K}"
13011 PRINT TAB(17)"{J}"
13012 RETURN
62000 PRINT "{clr 2°down}";TAB(15-LEN(PG$));PG$;"_BY_C.T._NADOVICH"
62010 PRINT "{down}.....CURSOR.#";NM$;"_COPYRIGHT_(C)_1981{down}"
62020 FOR I=1 TO 13:PRINT "{@@@}";:NEXT I:PRINT
62030 PRINT "{down}A_Tribute_To_Yakban..."
62040 PRINT "{2°down}PRESS_{rvs}RETURN{off}_TO_BEGIN"
62050 GOSUB 6500
62060 PRINT "{clr}":CLR
62070 QL=245:IF PEEK(50000) THEN QL=216
62080 GOTO 100
62090 GOSUB 6500:GOTO 60300

```



GAMES FOR FUN



MAD

By Glen Fisher

Here's your chance to be an author (well, at least a co-author). The PET will compose a story, and ask you for the proper words at important points in the story. There is one hitch, however. The PET won't tell you the story until after you've given it the words.

As the story is composed, the PET will ask you for the words that it needs, and you type them in at the keyboard. You'll need to know the difference between an adjective and an adverb for this one, as well as what a noun and a verb are, but that shouldn't be hard. After you have supplied the PET with the words it wants, it will display the story for all to see. Since the PET is not going to censor your words, you're on your honor.

If you want to change or add any stories, it can be done very simply. The variable NS is the number of stories (see line 150). The stories are stored in Data statements at line 10065. Each story begins with a graphic character "Q" (a shifted Q). This character must also be at the end

of the last story. A backslash means "go to the next line on the screen." Two backslashes will give you a blank line. The dictionary of word categories begins at line 10000. For example, when MAD finds a %A in the text of a story, it asks you to give it an adverb.

The following is a list of symbols used in stories for MAD:

Symbol	Represents
%A	adverb
%B	a part of the body (such as "hand")
%C	calamity (or disaster)
%D	disease
%E	exclamation
%J	adjective
%N	noun
%P	plural noun
%U	number
%V	verb
%Y	animal
%Z	a person's name

```
0 CLR:POKE 59468,12
6 REM MAD COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81 GF
90 PG$="MAD":NM$="2":GOSUB 62000
100 DIM P$(26)
105 DIM M$(100)
110 FOR I=0 TO 25:READ P$(I):NEXT I
150 NS=5:DIM USD(NS)
160 FOR I=1 TO NS:USD(I)=I:NEXT I
200 SC=NS:REM RESTART
300 PRINT "{clr 3°down}"
310 IF SC=0 THEN 160
320 RESTORE:FOR I=1 TO 26:READ T$:NEXT I
330 PRINT "PLEASE TYPE IN THE FOLLOWING:{down}"
340 IF SC>(NS-3) THEN 400
350 FOR I=1 TO NS:SN=I
355 IF USD(SN)>0 THEN 365
360 NEXT I
365 USD(SN)=-1:GOTO 406
400 I=INT(RND(1)*SC)+1
401 SN=USD(I)
402 IF USD(I)<1 THEN 400
```

```

403 USD(I)=-1
406 SC=SC-1
410 I=0
500 READ S$:IF LEFT$(S$,1)<>"{Q}" THEN 500
510 I=I+1:IF I<>SN THEN 500
520 N=0
530 S$=MID$(S$,2,255)
1000 IF S$="" THEN M$(N)=M$:N=N+1:M$="":READ S$:IF LEFT$(S$,1)="{Q}"
    THEN 2000
1010 C$=LEFT$(S$,1):S$=MID$(S$,2,255)
1020 IF C$<>"%" THEN M$=M$+C$:GOTO 1000
1025 IF S$="" THEN 1000
1030 C$=LEFT$(S$,1):S$=MID$(S$,2,255)
1040 T=ASC(C$)-ASC("A")
1050 PRINT :PRINT "A";
1060 FOR I=1 TO 5
1070 IF LEFT$(P$(T),1)=MID$("AEIOU",I,1) THEN PRINT "N";
1080 NEXT I
1100 PRINT "._";P$(T);":._";
1110 GOSUB 60000:T$=IN$:IF IN$="" THEN 1110
1120 M$=M$+T$:GOTO 1000
2000 I=-1:PRINT "{clr}";
2003 N=N-1:M$(N)=M$(N)+"._"
2005 I=I+1:IF I>N THEN 3000
2007 M$=M$(I)
2010 IF M$="" THEN 2005
2020 C$=LEFT$(M$,1):M$=MID$(M$,2,255)
2030 IF C$<>"._" AND C$<>"\" THEN W$=W$+C$:GOTO 2010
2050 IF LEN(L$)+LEN(W$)+1>39 THEN PRINT L$:L$=""
2060 L$=L$+"._"+W$:
2065 IF C$="\" THEN PRINT L$:L$=""
2070 W$="":GOTO 2010
3000 PRINT L$:L$=""
3010 PRINT "{down}._WANT._TO._TRY._ANOTHER?._";:GOSUB 60000:RJ$=IN$
3020 IF LEFT$(RJ$,1)="N" THEN 3500
3025 GOTO 300
3500 PRINT "{clr 4°down}THANKS._FOR._PLAYING.":END
10000 DATA ADVERB,"PART._OF._YOUR._BODY"
10004 REM C=CALAMITY
10005 DATA DISASTER,DISEASE,EXCLAMATION
10010 DATA F,G,H,I,ADJECTIVE
10020 DATA K,L,M,NOUN,O
10030 DATA "PLURAL._NOUN",Q,R,S,T
10040 DATA NUMBER,VERB,W,X,ANIMAL
10050 DATA "NAME._OF._A._PERSON._YOU._KNOW"
10065 DATA "{Q}DEAR._SIRS:\""
10070 DATA
    "I._RECENTLY._PURCHASED._A._%J._COMPUTER._FROM._A._%J._STORE._.I._GOT._IT._FOR._"
10075 DATA "%Z,._WHO._JUST._LOVES._ALL._SORTS._OF._%J._GADGETS._."
10080 DATA "._WHEN._I._BROUGHT._THE._%J._COMPUTER._TO._MY._%N,._IT._WAS._SO._%J"
10090 DATA "._THAT._IT._MADE._MY._%B._ACHE._.I._READ._YOUR._%J._MANUAL,._"

```

```

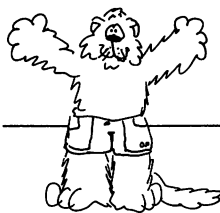
10100 DATA "BUT.IT.DIDN'T.HELP.ME.UNDERSTAND.THE.%J.COMPUTER.."
10110 DATA "I.THINK.THIS.COULD.LEAD.TO.A.SERIOUS.CASE.OF.%D."
10120 DATA
    "EVERY.%J.PROGRAM.I.WRITE.SEEMS.TO.BE.%J..CAN.I.EXCHANGE."
10130 DATA "THIS.%J.COMPUTER.FOR.ONE.THAT.WILL.LET.ME.%V"
10140 DATA "PROGRAMS,,AND.THAT.WON'T.MAKE.MY.%B.SORE?"
10150 DATA
    "\\P.S...I.HAVE.%U.OTHER.%P,,AND.THEY.ALL.WORK.BETTER.THAN."
10160 DATA "YOUR.%J.LITTLE.%Y!"
11210 DATA
    "{Q}ONCE.UPON.A.TIME,,A.%J.WOMAN.FELL.IN.LOVE.WITH.HER.%J.COMPUTER.."
11220 DATA
    "SHE.WOULD.BEGIN.EACH.DAY.BY.KISSING.ITS.%N.AND.%A.CARESSING.ITS.%P."
11230 DATA
    "THEN.ONE.DAY,,TRAGEDY.STUCK:.SHE.AWOKE,,AND.FOUND.HER.COMPUTER."
11240 DATA "DESTROYED.BY.A.%C.%E!.SHE.SAID,,HOW.CAN.I.EVER.BE."
11250 DATA "%J.AGAIN?BUT.THEN.A."
11260 DATA
    "WONDERFUL.THING.CAME.TO.PASS:.%A,,A.BRAND.NEW.%J.COMPUTER.APEARED"
11270 DATA "ON.HER.DESK..IT.HAD.A.%J.USERS'S.%N,,AND.IT.WORKED."
11280 DATA "%A!.'%E!.THE.%N.LIKES.ME!',"SHE.EXCLAIMED.\\
11310 DATA "{Q}\\HOW.TO.GET.AHEAD...\\
11320 DATA "FIRST,,YOU.MUST.ALWAYS.TRY.TO.%V.YOUR.BOSS.."
11330 DATA "GO.TO.WORK.%A,,AND.ALWAYS.EAT.YOUR.%N.AT.YOUR.%N.."
11340 DATA "BE.%J.WHEN.YOU.ANSWER.THE.%N,,AND.WHEN.YOU.ARE."
11350 DATA "ASKED.TO.%V.A.%J.JOB,,BE.SURE.AND.GIVE.IT.YOUR."
11360 DATA "BEST.%N.\\REMEMBER,,THE.CUSTOMER.IS.ALWAYS.%J!"
11370 DATA "{Q}\\RECENT.EMPLOYEE.INJURIES:\\"
11380 DATA
    "AN.ELECTRICAL.MAINTENANCE.MAN.TOUCHED.A.%N.TO.A.METAL.%N.."
11390 DATA "THE.EXPLOSION.HURT.HIS.%B.\\
11400 DATA "A.PROFESSOR.WAS.HIT.BY.A.%N.AND.SUFFERED."
11410 DATA "BRUISES.AND.CUTS.ON.HER.%B.\\
11420 DATA "A.FOOD.SERVICE.WORKER.FELL.WHILE.MOPPING.THE.%N,."
11430 DATA "INJURING.HER.LEGS.\\
11440 DATA
    "A.COMPUTER.PROGRAMMER.FELL.ASLEEP.AT.HER.PET(TM).COMPUTER"
11450 DATA "A.%C.OCCURRED.WHILE.SHE.SLEPT,"
11460 DATA "CAUSING.HER.%J.COMPUTER.TO.FALL.ON.HER.%B.\\
11470 DATA "{Q}CLASSIFIED.ADS:\\"
11480 DATA "FREE.%N.LESSONS.IN.EXCHANGE.FOR.KEEPING.%N."
11490 DATA "FREE.OF.%N..(766-2337)\\
11500 DATA "VOLUNTEERS.NEEDED.FOR.RESEARCH.PROJECT..MUST.NOT."
11510 DATA "BE.AFRAID.OF.%P,,%P,,OR.A.%J.%N..FIND.OUT.MORE."
11520 DATA "BY.CALLING.%Z.AT.968-9905\\
11530 DATA "SICKLY.%Y.FOR.SALE..(976-1880)\\
11540 DATA "{Q}"
60000 IN$="":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
    ZT=TI+15

```

```

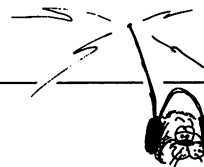
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "{left}";:
      GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
      Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 GOSUB 60500:PRINT "{down}PRESS_{rvs}RETURN{off}_{TO}";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTIUNE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN CLR:PRINT "{clr}";:GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_{CURSOR}_#";NM$;
      "_{(C)}_1979_{PAGE}";PN;"{down}"
60270 GOSUB 60500:RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;"_{BY}_GLEN_FISHER"
62010 PRINT "{down}_{CURSOR}_#2_{COPYRIGHT}_{(C)}_1978"
62015 GOSUB 60500
62020 PRINT "{down}WILD_AND_CRAZY_STORIES"
62030 PRINT "{2°down}WANT_INSTRUCTIONS?";
62040 GOSUB 60000:IF IN$<>"Y" THEN 100
62100 LP=1:GOSUB 60200
62105 PRINT "FILL_IN_THE_BLANKS_TO_CREATE_A_FUNNY"
62110 PRINT "STORY."
62120 PRINT "{down}THE_COMPUTER_ASKS_FOR_WORDS,THEN_USES"
62130 PRINT "THOSE_WORDS_TO_FILL_IN_THE_BLANKS_TO_A"
62135 PRINT "STORY."
62140 PRINT "{down}HAVE_FUN!_(SEE_PRINTED_DIRECTIONS"
62150 PRINT "FOR_INSTRUCTIONS_ABOUT_ADDING_YOUR_OWN"
62160 PRINT "STORIES_TO_THIS_PROGRAM)."
62165 GOSUB 60200

```



QUIX

By Glen Fisher

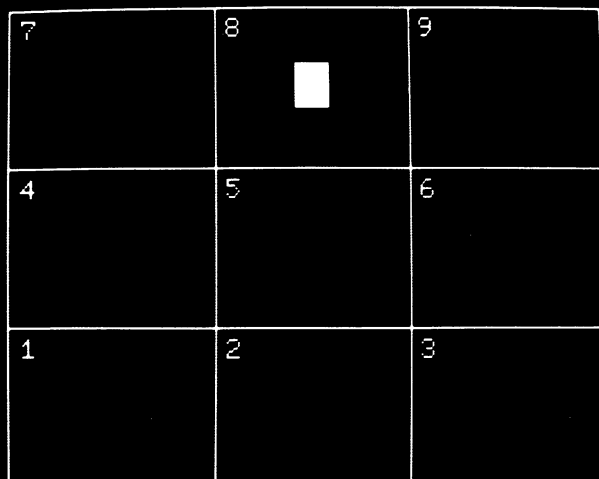


On a large 3×3 square, the PET will show a pattern of lights and sound. Your mission is to repeat the pattern, light for light and sound for sound. If you succeed, the next one will be a little harder (either the pattern will be shown faster, or it will be longer). If you fail, the PET will come back with a slower or shorter pattern.

The 3×3 square on the screen corresponds to the 3×3 numeric keypad (the keys [1] through [9]). Each square on the screen contains the number of the corresponding key. To repeat the pattern, press the keys on the keypad in the same order that the squares were lit in the pattern. You

may take as long as you like to repeat the pattern, but the longer you take, the more likely you are to forget the pattern.

Unlike other popular memory-recall games, the patterns presented by QUIX are random sequences, which makes QUIX quite a bit more challenging to play. The only "tuning" you can do to QUIX is at line 100. The variables F1 and F2 control how often the PET makes the pattern faster (or slower), rather than longer (or shorter). The speed is changed F1/F2 of the time. (for example, if F1=1 and F2=4, the speed will change one-fourth of the time.)



THIS WAS: 26/60 SECS., LENGTH 3
BEST: 27/60 SECS., LENGTH 3

```
0 CLR:POKE 59468,12
6 REM QUIX! COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81 RJ
90 PG$="QUIX!":NM$="3":GOSUB 62000
100 F1=1:F2=2:REM PICKS DISPLAY TIME CHANGE F1/F2 OF THE TIME
101 PRINT CHR$(142)
400 FOR I=0 TO 9:READ PCH(I):NEXT I
410 DATA 0,237,211,188,177,157,140,124,117,104
500 W$="{rvs}...{2°left down}...{off}":D$="...{2°left down}..."
510 DIM CM$(9):CM$="{home 4°down 7°right}"
520 FOR I=0 TO 2
530 V$=MID$("{14°down}",7*(2-I)+1,255)
```

```

540 FOR J=1 TO 3
550 H$=MID$("{33°right}",11*J,255)
560 CM$(10-(3*I+J))=V$+H$
570 NEXT J
580 NEXT I
600 GOSUB 8000
610 DN$="{home 23°down}"
620 PRINT DN$;"{4°right}THIS WAS:.... /60 SECS., LENGTH"
630 PRINT "{4°right}BEST:..... /60 SECS., LENGTH{home}"
700 LN=1:DL=30
710 BL=0:BD=1E8
720 GOTO 2450
1000 SQ$="":T=0:WR=0
1010 GOSUB 5000
1100 FOR I=1 TO LN
1110 J=INT(RND(1)*9+1):IF J=T THEN 1110
1120 :SQ$=SQ$+MID$("123456789",J,1):T=J
1130 NEXT I
1140 GOSUB 7000
2000 T=1:GOSUB 5000
2100 GET T$:IF T$<>" THEN 2105
2101 IF TI>OF THEN GOSUB 6100:OF=1E8
2102 GOTO 2100
2105 IF T$="_" OR T$="Q" THEN GOSUB 6100:GOTO 4000
2106 IF T$=CHR$(13) THEN GOSUB 6100:GOTO 2300
2110 IF T$<"1" OR T$>"9" THEN 2100
2120 GOSUB 7600:SQ=VAL(T$):GOSUB 7500
2125 GOSUB 6000:GOSUB 6200:OF=TI+45
2130 IF T$<>MID$(SQ$,T,1) THEN 2200
2140 IF T=LN THEN 2250
2150 T=T+1:GOTO 2100
2200 T$="{rvs}WRONG!!!{off}":CH=-1:WR=1:GOTO 2400
2250 T$="CORRECT!":CH=1:GOTO 2400
2300 T$="STARTING OVER":CH=0
2310 DL=30:LN=1
2400 GOSUB 7600:GOSUB 2700:GOSUB 3000
2405 GOSUB 6100
2410 IF RND(1)*F2>=F1 THEN 2420
2415 IF DL-CH>0 THEN DL=DL-CH:GOTO 2430
2420 IF LN+CH>0 THEN LN=LN+CH
2430 GOSUB 5000:D=DL:DL=90:GOSUB 7400:DL=D
2436 GET I$:IF I$<>" THEN 2436
2440 GOSUB 2750:GET I$:IF I$="" THEN 1000
2445 IF I$="_" THEN 4000
2450 T$="PRESS_{rvs}RETURN{off}":GOSUB 2700
2460 GET I$:IF I$="" THEN 2460
2465 IF I$="_" THEN 4000
2470 GOSUB 2750:GOTO 1000
2700 PRINT "{home}":PRINT "{up}";TAB(20-LEN(T$)/2);T$
2710 RETURN
2750 PRINT "{home}":PRINT "{up}";TAB(20-LEN(T$)/2);

```

```

2760 PRINT LEFT$(".....",LEN(T$))
2770 RETURN
3000 PRINT DN$;"{13°right}";
3010 PRINT RIGHT$("...."+STR$(DL),3);
3020 PRINT "{17°right}";LN
3030 IF WR OR DL>BD OR LN<BL THEN RETURN
3040 PRINT "{13°right}";
3050 BD=DL:BL=LN
3060 PRINT RIGHT$("...."+STR$(BD),3);
3070 PRINT "{17°right}";BL;
3900 RETURN
4000 PRINT DN$;"{down 4°ins}.....{13°right}";
4010 T$=CHR$(20):PRINT T$;T$;T$;T$;"{2°up}"
4020 END
5000 GET Z$:IF Z$<>" THEN 5000
5010 RETURN
6000 :
6010 POKE 59467,16
6020 POKE 59464,0
6030 POKE 59466,15
6040 RETURN
6100 :
6110 POKE 59467,0
6120 POKE 59466,0
6130 RETURN
6200 :
6210 POKE 59464,PCH(SQ)
6220 RETURN
7000 GOSUB 6000
7005 FOR I=1 TO LEN(SQ$)
7010 SQ=VAL(MID$(SQ$,I,1))
7020 GOSUB 7500
7025 GOSUB 6200
7030 GOSUB 7400
7050 GOSUB 7600
7060 NEXT I
7065 GOSUB 6100
7070 RETURN
7400 TM=TI+DL
7410 IF TI<TM THEN 7410
7420 RETURN
7500 GOSUB 7700:PRINT W$;:RETURN
7600 GOSUB 7700:PRINT D$;:RETURN
7700 PRINT CM$;CM$(SQ);:RETURN
8000 L$="@@@@@@@@@@":S$="....."
8010 PRINT "{clr down}...{0}";L$;"{2}";L$;"{2}";L$;"{.}"
8020 T$="...{+}"+S$+"{+}"+S$+"{+}"+S$+"{+}"
8030 FOR I=1 TO 6:PRINT T$:NEXT I
8040 PRINT "...{+}";L$;"{+}";L$;"{+}";L$;"{+}"
8050 FOR I=1 TO 6:PRINT T$:NEXT I
8060 PRINT "...{+}";L$;"{+}";L$;"{+}";L$;"{+}"

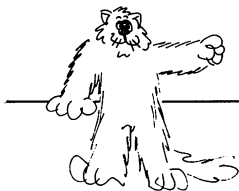
```



```

8070 FOR I=1 TO 6:PRINT T$:NEXT I
8080 PRINT "._._{-}";L$;"{1}";L$;"{1}";L$;"{=}"
8090 H$="{10°right}":V$="{6°down}"
8100 PRINT "{home 2°down}";"{4°right}7";H$;"8";H$;"9"
8110 PRINT V$;"{4°right}4";H$;"5";H$;"6"
8120 PRINT V$;"{4°right}1";H$;"2";H$;"3"
8130 RETURN
60000 IN$="._":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("._{&}",ZC,1);"{left}";:ZC=3-ZC:ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "._{left}";:
GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 GOSUB 60500:PRINT "{down}PRESS._{rvs}RETURN{off}._TO._";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTIUNE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN CLR:PRINT "{clr}";:GOTO 100
60260 PN=PN+1:PRINT "{clr}";PG$;"._-._CURSOR._#";NM$;
"._.(C)._1981._._.PAGE";PN;"{down}"
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;"._.BY._GLEN._FISHER"
62005 PRINT "{down}._._._CURSOR._#3._.COPYRIGHT._(C)._1981"
62006 GOSUB 60500
62010 PRINT "{down}HOW._GOOD._IS._YOUR._MEMORY?"
62020 PRINT "{2°down}WANT._DIRECTIONS?._";:GOSUB 60000
62030 IF LEFT$(IN$,1)<>"Y" THEN 100
62100 LP=1:GOSUB 60200
62160 PRINT "THIS._IS._A._GAME._THAT._TESTS._YOUR._MEMORY."
62170 PRINT "THE._PET._WILL._DISPLAY._A._SEQUENCE,._AND"
62180 PRINT "YOU._TRY._TO._REPEAT._THE._SEQUENCE._AS._YOU"
62190 PRINT "GET._BETTER,._THE._SEQUENCE._GETS._LONGER,"
62200 PRINT "AND._IS._FLASHED._MORE._QUICKLY."
62202 PRINT "{down}USE._THE._NUMERIC._KEYS._TO._ENTER._YOUR"
62204 PRINT "PATTERN._.WHEN._YOU._PRESS._AN._INCORRECT{._}"
62206 PRINT "KEY,._THE._PROGRAM._WILL._SAY{._rvs}WRONG!!!{off},_"
62208 PRINT "THEN._SHOW._YOU._A._NEW._SEQUENCE."
62210 PRINT "{down}YOU._CAN._START._OVER._BY._PRESSING._THE"
62220 PRINT "{rvs}RETURN{off}._KEY._.PRESS._{rvs}Q{off}._TO._END"
62230 PRINT "THE._GAME."
62240 GOSUB 60200

```



BOP

By Glen Fisher

Before calculators and computers were invented, the Koreans invented the “Chisanbop” method of counting. The fingers on the right hand are each worth one, and the thumb is worth five. The fingers on the left hand are worth ten each, and the thumb 50. To show a number with your fingers, extend the fingers that add up to the number you want.

In our version, we’ve trapped someone inside the PET, forcing him to press his hands against the glass. He is very obedient and will extend or fold over whichever fingers he’s told to, either by you or the PET.

When you start, you’ll be presented with five choices. The first four will be explained below. The fifth just lets you out of the program (and lets the little man rest). Once you’ve made a choice, you can get back to the list of choices by pressing [RETURN].

The first choice (“SET UP FINGERS”) instructs you to set the man’s fingers to the number the PET tells you. If you get it wrong, the PET will tell you to what number you set the man’s fingers. If it’s right, you’ll be congratulated. To set his fingers, you’ll see a small pointer on the screen which points to one of his fingers. You can move the pointer with the [<] and [>] keys ([<] moves the pointer left,

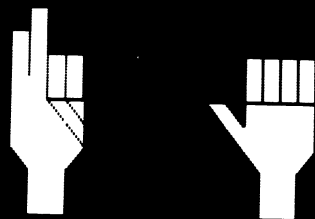
and [>] moves it right). When the pointer is pointing to the finger you want, press [RVS], which reverses the position of the finger (if it’s extended it will fold over, and if it’s folded it’ll be extended). When the fingers are set the way you think they should be, press [RETURN]. The PET will then tell you how well you did. When you’ve tired of setting up numbers, just press [RETURN] without doing anything, and you’ll go back to the list of choices.

The second choice (“READ THEM”) arranges the fingers, and has you say what number is shown. If you misread the fingers, the PET will tell you what number was on them. Again, when you tire of this, just press [RETURN], and you’ll be back at the list of choices.

The third choice (“WATCH THEM”) lets you pick numbers, which the man will then put onto his fingers. Obviously, this is an excellent way to learn the Chisanbop system. As always, [RETURN] will get you back to the list of choices.

The fourth choice (“COUNT TO 99”) puts the man through his paces as he counts from zero to 99 with his fingers. When he gets tiresome, pressing [RETURN] will stop him, and return you to the list of choices.

SET UP THE NUMBER 74.



```

0 CLR:POKE 59468,12
6 REM BOP COPYRIGHT(C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81 GTF
50 PG$="BOP":GOSUB 62000
100 CLR:PRINT "{clr}";CHR$(142):DIM RC(9,1),UP(9)
110 FOR I=0 TO 9:READ RC(I,0),RC(I,1):NEXT I
120 DATA 7,9,5,12,5,13,7,16,12,17
130 DATA 12,21,7,22,5,25,5,26,7,29
140 PT$="{MG|N@MBHN}"
160 DN$="{left 24°down}"
190 ZER=0:RN=(RC(4,0)+RC(5,0))/2+1:CN=(RC(4,1)+RC(5,1))/2
210 CR$=CHR$(13)
500 PRINT "{clr}";
505 PRINT "{down}DO YOU WANT TO:"
510 PRINT " 1) SET UP FINGERS"
515 PRINT " 2) READ THEM"
520 PRINT " 3) WATCH THEM"
525 PRINT " 4) COUNT TO 99"
526 PRINT " 5) QUIT"
530 PRINT "ENTER YOUR CHOICE: ";:GOSUB 60000
540 IF IN$="1" THEN GOSUB 9000:GOTO 1000
545 IF IN$="2" THEN GOSUB 9000:GOTO 3500
550 IF IN$="3" THEN GOSUB 9000:GOTO 3000
555 IF IN$="4" THEN GOSUB 9000:GOTO 3700
556 IF IN$<>"5" THEN GOTO 560
557 PRINT "{clr down}":END
560 PRINT "{clr}PLEASE TYPE '1','2','3','4', OR '5'.{down}"
570 GOTO 510
1000 NUM=INT(RND(1)*99)+1:RJ=0
1010 PRINT "{home}SET UP THE NUMBER";NUM;"{left}."
1030 NF=0
1050 PRINT "{home}";SPC(1+RC(NF,1));LEFT$(DN$,1+RC(NF,0));
1060 PRINT MID$(PT$,1+NF,1);"{left}";
1100 GET T$:IF T$="" THEN 1100
1110 PRINT " {left}";
1120 IF T$="4" OR T$="<" THEN NF=NF-1:IF NF<0 THEN NF=9
1130 IF T$="6" OR T$=">" THEN NF=NF+1:IF NF>9 THEN NF=0
1140 IF T$="{rvs}" THEN GOSUB 4000
1150 IF T$=CHR$(13) AND RJ=1 THEN 1200
1151 IF T$=CHR$(13) AND RJ=0 THEN 500
1152 RJ=1
1155 IF T$=" " THEN 500
1160 GOTO 1050
1200 N=0:FOR I=3 TO 0 STEP -1
1210 IF UP(I)=ZER THEN 1230
1220 N=N+10:NEXT I
1230 IF I<=0 THEN 1270
1240 FOR I=I TO 0 STEP -1
1250 IF UP(I)<>ZER THEN 1900
1260 NEXT I
1270 N=N+50*ABS(UP(4)<>ZER)

```

```

1280 FOR I=6 TO 9
1290 IF UP(I)=ZER THEN 1310
1300 N=N+1:NEXT I
1310 IF I>=9 THEN 1350
1320 FOR I=I TO 9
1330 IF UP(I)<>ZER THEN 1900
1340 NEXT I
1350 N=N+5*ABS(UP(5)<>ZER)
1360 IF N=NUM THEN PRINT "{home down}RIGHT!";:GOTO 1500
1370 PRINT "{home down}NO.";:GOTO 1500
1500 PRINT " _THE_NUMBER_ON_MY_FINGERS_IS";N;"{left}."
1510 FOR I=1 TO 2000:NEXT I
1520 GOSUB 5500
1530 GOTO 1000
1900 PRINT "{home down}THAT_ISN'T_EVEN_A_NUMBER!":GOTO 1510
2000 T=INT(RND(1)*1024)
2010 FOR NF=0 TO 9
2030 UP=1 AND T:T=INT(T/2)
2040 IF UP(NF)<>UP THEN GOSUB 4000
2050 NEXT NF
2060 GOTO 2000
2990 FOR I=1 TO 1000:NEXT I
3000 GOSUB 7000
3010 NUM=NM:IF NM<0 THEN 500
3020 GOSUB 6000
3030 FOR NF=0 TO 9
3050 IF UP(NF)<>F(NF) THEN GOSUB 4000
3060 NEXT NF
3070 PRINT "{home}";SPC(CN-(NM>9));LEFT$(DN$,RN);NUM;"{left}_"
3080 GOTO 3000
3500 NUM=INT(RND(1)*100):NM=NUM:GOSUB 6000
3510 FOR NF=0 TO 9
3530 IF UP(NF)<>F(NF) THEN GOSUB 4000
3540 NEXT NF
3550 GOSUB 7000:IF NM<0 THEN 500
3560 IF NM=NUM THEN PRINT "{home down}RIGHT!";:GOTO 3580
3570 PRINT "{home down}NO.";
3580 PRINT " _THE_NUMBER_IS";NUM;"{left}."
3590 FOR I=1 TO 2000:NEXT I
3600 GOTO 3500
3700 FOR NUM=0 TO 99:NM=NUM:GOSUB 6000
3701 GET T$:IF T$<>" THEN 500
3705 PRINT "{home}";SPC(CN-(NM>9));LEFT$(DN$,RN);NUM;"{left}_"
3710 FOR NF=0 TO 9
3720 PRINT "{home}";SPC(1+RC(NF,1));LEFT$(DN$,1+RC(NF,0));
3730 IF UP(NF)<>F(NF) THEN GOSUB 4000
3740 NEXT NF
3750 FOR I=1 TO 500:NEXT I
3760 NEXT NUM
3770 GOTO 500
4000 PRINT "{home}";SPC(1+RC(NF,1));LEFT$(DN$,1+RC(NF,0));

```

```

4010 ON NF+1 GOSUB 5000,5010,5020,5030,5040,5050,5060,5070,5080,5090
4020 PRINT "{off}";:RETURN
5000 PRINT "{2°right down}";:IF UP(0) THEN PRINT "_{left 2°down rvs
:}";:GOTO 5003
5002 PRINT "{rvs ' left 2°down}";MID$("{' }_",1+UP(1),1);
5003 PRINT "{left 3°up 2°left}";:UP(0)=1-UP(0)
5005 RETURN
5010 PRINT "{2°down}";:IF UP(1) THEN PRINT "_{left down}_ {left
2°down left rvs}";MID$("{: }_",1+UP(0),1);"{:}";:GOTO 5013
5012 PRINT "{rvs ' left down ' left 2°down left}";MID$("{: }_",
1+UP(0),1);MID$("{' }_",1+UP(2),1);
5013 PRINT "{left 5°up}";:UP(1)=1-UP(1)
5015 RETURN
5020 PRINT "{2°down}";:IF UP(2) THEN PRINT "_{left down}_ {left
2°down left rvs}";MID$("{: }_",1+UP(1),1);"{:}";:GOTO 5023
5022 PRINT "{rvs ' left down ' left 2°down left}";MID$("{: }_",
1+UP(1),1);MID$("{: $' }_",1+UP(3)+2*UP(4),1);
5023 PRINT "{left 5°up}";:UP(2)=1-UP(2)
5025 RETURN
5030 PRINT "{2°left down}";
5031 IF UP(3) THEN PRINT "_{left 2°down left rvs}";MID$("{: }_",
1+(UP(2) AND UP(4)),1);"{:}";:GOTO 5033
5032 PRINT "{rvs ' left 2°down left}";MID$("{: $: }_",1+UP(2)+2*UP(4),
1);"{' }";
5033 PRINT "{right 3°up}";:UP(3)=1-UP(3)
5035 RETURN
5040 PRINT "{4°left 2°up rvs}";
5041 IF UP(4) THEN PRINT MID$("{: $' }_",1+(UP(2) AND UP(3)),1);"{left
down MM left down M off}_ {left up}_";:GOTO 5043
5042 PRINT MID$("{: ': }_",1+UP(2)+2*UP(3),1);"{left down}_ { ' left
down}_ {off ) left up rvs ) off )}";
5043 PRINT "{down}";:UP(4)=1-UP(4)
5045 RETURN
5050 PRINT "{up right}";
5051 IF UP(5) THEN PRINT "__{left down}_ {rvs N left up NN left up}";
MID$("{L$ }_",1+(UP(6) AND UP(7)),1);:GOTO 5053
5052 PRINT "{< rvs < off left down < rvs}_ {left up %}_ {left up}";
MID$("{LL% }_",1+UP(6)+2*UP(7),1);
5053 PRINT "{2°down 5°left}";:UP(5)=1-UP(5)
5055 RETURN
5060 PRINT "{2°right down}";
5061 IF UP(6) THEN PRINT "_{left 2°down rvs L}";MID$("{L% }_",1+(UP(5)
AND UP(7)),1);:GOTO 5063
5062 PRINT "{rvs % left 2°down %}";MID$("{LL$ }_",1+UP(5)+2*UP(7),1);
5063 PRINT "{2°left 3°up 2°left}";:UP(6)=1-UP(6)
5065 RETURN
5070 PRINT "{2°down}";
5071 IF UP(7) THEN PRINT "_{left down}_ {left 2°down rvs L}";
MID$("{L% }_",1+UP(8),1);:GOTO 5073
5072 PRINT "{rvs % left down % left 2°down}";MID$("{L% $' }_",
1+UP(5)+2*UP(6),1);MID$("{L }_",1+UP(8),1);

```

```

5073 PRINT "{2°left 5°up}";:UP(7)=1-UP(7)
5075 RETURN
5080 PRINT "{2°down}";
5081 IF UP(8) THEN PRINT "_{left down}_{left 2°down rvs L}";
      MID$("{L}",1+UP(9),1);:GOTO 5083
5082 PRINT "{rvs % left down % left 2°down}";MID$("{%}_",1+UP(7),1);
      MID$("{L}_",1+UP(9),1);
5083 PRINT "{2°left 5°up}";:UP(8)=1-UP(8)
5085 RETURN
5090 PRINT "{2°left down}";
5091 IF UP(9) THEN PRINT "_{left 2°down rvs L}";:GOTO 5093
5092 PRINT "{rvs % left 2°down}";MID$("{%}_",1+UP(8),1);
5093 PRINT "{right 3°up}";:UP(9)=1-UP(9)
5095 RETURN
5500 PRINT "{home off}";
5510 FOR II=1 TO 3
5520 PRINT "....."
5530 NEXT II
5540 PRINT "{home}";:RETURN
6000 REM
6010 FOR I=0 TO 9:F(I)=ZER:NEXT I
6020 T=INT(NM/5):N=NM-5*T:NM=T
6030 IF N>0 THEN FOR I=6 TO 5+N:F(I)=1-F(I):NEXT I
6040 T=INT(NM/2):N=NM-2*T:NM=T
6050 IF N THEN F(5)=1-F(5)
6060 T=INT(NM/5):N=NM-5*T:NM=T
6070 IF N>0 THEN FOR I=3 TO 4-N STEP -1:F(I)=1-F(I):NEXT I
6080 T=INT(NM/2):N=NM-2*T:NM=T
6090 IF N THEN F(4)=1-F(4)
6100 RETURN
7000 GOSUB 5500
7100 PRINT "{home}NUMBER?_";:GOSUB 60000
7105 IF IN$="" THEN NM=-1:RETURN
7110 IF IN$="" THEN 7300
7120 FOR I=1 TO LEN(IN$)
7130 :T$=MID$(IN$,I,1):IF T$>="0" AND T$<="9" OR T$="-" THEN 7160
7140 IF T$="." THEN PRINT "{home down}NO_FRACTIONS,_PLEASE.":
      GOTO 7310
7150 GOTO 7300
7160 NEXT I
7170 NM=VAL(IN$)
7180 IF NM<0 THEN PRINT "{home down}TOO_SMALL.":GOTO 7310
7190 IF NM>99 THEN PRINT "{home down}TOO_BIG.":GOTO 7310
7200 RETURN
7300 PRINT "{home down}THAT'S_NOT_A_NUMBER."
7310 FOR I=1 TO 1000:NEXT I
7320 GOTO 7000
9000 PRINT "{clr 12°right 7°down}";
9010 PRINT "{rvs '' 3°left down '''' 4°left down ''''}";
9020 PRINT "{4°left down}...{' 4°left down}...{' ) off )}";
9030 PRINT "{6°left down rvs}.....{off ) 5°left down}";
9040 PRINT "{< rvs}...{off ) 3°left down rvs}...{2°left down}...";

```

```

9050 PRINT "{8°up 11°right}";
9060 PRINT "{rvs %% 3°left down %%% 4°left down %%%}";
9070 PRINT "{4°left down %}...{6°left down off < rvs <%}...";
9080 PRINT "{5°left down off < rvs}...{4°left down}";
9090 PRINT "{off < rvs}...{off ) 3°left down rvs}...{2°left
down}...{off}";
9100 FOR II=0 TO 9:UP(II)=1:NEXT II
9110 RETURN
60000 IN$="_":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>" " THEN 60070
60020 IF ZT<=TI THEN PRINT MID$("_{&}",ZC,1);"{left}";:ZC=3-ZC:
ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AND 127)<32 THEN PRINT "_{left}";:
GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (Z AND 127)<91 THEN
Z$=CHR$((Z+128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
NEXT Z:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN 60250
60210 GOSUB 60500:PRINT "{down}PRESS_{rvs}RETURN{off}_{TO}_";
60220 IF PN=LP THEN PRINT "BEGIN:":GOTO 60240
60230 PRINT "CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN 100
60260 PN=PN+1:PRINT "{clr}";PG$;"_-_-CURSOR_#";NM$;
"_(C)_1981...PAGE";PN;"{down}"
60270 GOSUB 60500:RETURN
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;"_BY_GLEN_FISHER"
62005 PRINT "{down}...CURSOR_#4_COPYRIGHT_(C)_1981"
62006 GOSUB 60500
62010 PRINT "{down}A_GAME_THAT_TEACHES_CHISANBOP"
62020 PRINT "{2°down}WANT_DIRECTIONS?_";:GOSUB 60000
62030 IF IN$<>"Y" THEN 100
62040 LP=3:NM$="4":PG$="BOP":GOSUB 60200
62060 PRINT "THIS_GAME_PPRESENTS_THE_KOREAN_METHOD"
62070 PRINT "{down}OF_COUNTING_CALLED_CHISANBOP."
62090 PRINT "{down}THIS_TECHNIQUE_MAKES_IT_POSSIBLE"
62110 PRINT "{down}TO_COUNT_TO_{_}99_ON_YOUR_FINGERS."
62170 GOSUB 60200
62185 PRINT "CHISANBOP_RULES:"
62190 PRINT "{down}FINGERS_ON_YOUR_RIGHT_HAND_ARE_WORTH_1"
62200 PRINT "{down}YOUR_RIGHT_THUMB_IS_WORTH_{_}5"
62210 PRINT "{down}FINGERS_ON_YOUR_LEFT_HAND_ARE_WORTH_{_}10"

```

```

62220 PRINT "{down}YOUR LEFT THUMB IS WORTH{ }50"
62250 PRINT "{down}PRESS YOUR FINGERS AGAINST A TABLE"
62260 PRINT "{down}TO{ } 'MAKE A NUMBER' ...FOR EXAMPLE,"
62270 PRINT "{down}FOR THE NUMBER{ }6,{ }PRESS YOUR{ }"
62280 PRINT "{down}RIGHT THUMB AND FOREFINGER ON THE TABLE."
62290 GOSUB 60200
62300 PRINT "TO SET THE FINGERS IN THIS GAME:"
62310 PRINT "{down}PRESS{ } '>' TO GO RIGHT,"
62320 PRINT "{down}PRESS '<' TO GO LEFT."
62330 PRINT "{down}TO CHANGE A FINGER, USE THE{ } 'RVS' KEY."
62335 PRINT "{down}AFTER YOU HAVE 'SET' THE FINGERS,"
62337 PRINT "{down}PRESS {rvs} RETURN{off} TO ENTER YOUR ANSWER"
62340 PRINT "{2°down}THERE ARE SEVERAL OPTIONS IN THIS GAME."
62350 PRINT "{down}YOU CAN ALWAYS GO BACK TO THE MAIN MENU"
62360 PRINT "{down}BY PRESSING RETURN."
62400 GOSUB 60200

```




FACE

By Glen Fisher

FACE allows you and the PET to design different faces on the screen. By combining a variety of available eyebrows, eyes, noses, and mouths, you can create some very interesting expressions (as well as a number of rather dull ones). If you need some creative inspiration, FACE also will display some of its efforts for you.

FACE will show you a list of choices from which you can choose to create your own, or see some of the faces that the PET created. The first choice is random faces. The PET randomly puts faces all over the screen (which leads to some interesting designs) until you press [RETURN]. The second choice is a "rogue's gallery," which is a neatly displayed gallery of faces, and different every time. The third choice presented is a "do-it-yourself," which allows you to design your own

faces. The last choice is to end the game. You can always get back to the list of choices by pressing [RETURN]

When you design your own face, you must first choose a pair of eyebrows. This tends to set the character of the face. For instance, if you choose angry eyebrows, it can't help but scowl, and if you choose laughing eyebrows, then even a frown becomes comical. After you have selected some eyebrows, you are shown a set of eyes that go with the eyebrows. After you have decided upon the eyes, you must choose a nose and a mouth to go with them.

When you have chosen all the features, the face you designed will be displayed on the screen. To create another face, press [SPACE]. To return to the list of choices, press [RETURN]

```
0 CLR:POKE 59468,12
6 REM FACE COPYRIGHT (C) 1981 THE CODE WORKS
10 REM AS OF 10/24/81 RJ
90 PG$="FACE":GOSUB 62000
100 DEF FN R(X)=INT(RND(1)*X)+1
110 CR$=CHR$(13)
1000 READ AN:DIM A$(AN),AH(AN),AW(AN):FOR I=1 TO AN:READ AH(I),
    AW(I),A$(I):NEXT I
1010 READ BN:DIM B$(BN),BH(BN),BW(BN):FOR I=1 TO BN:READ BH(I),
    BW(I),B$(I):NEXT I
1020 READ CN:DIM C$(CN),CH(CN),CW(CN):FOR I=1 TO CN:READ CH(I),
    CW(I),C$(I):NEXT I
1030 READ DN:DIM D$(DN),DH(DN),DW(DN):FOR I=1 TO DN:READ DH(I),
    DW(I),D$(I):NEXT I
1040 READ EN:DIM E$(EN),EH(EN),EW(EN):FOR I=1 TO EN:READ EH(I),
    EW(I),E$(I):NEXT I
1050 DIM NN(5):NN(1)=AN:NN(2)=BN:NN(3)=CN:NN(4)=DN:NN(5)=EN
1060 NN=0:FOR I=1 TO 5:IF NN(I)>NN THEN NN=NN(I)
1070 NEXT I
1080 DIM SV(NN)
1500 PRINT "{clr 4°down}HERE'S WHAT'S AVAILABLE:"
1510 PRINT "{down}_1)_RANDOM_FACES"
1520 PRINT "{down}_2)_A 'ROGUE'S GALLERY'"
1530 PRINT "{down}_3)_DO-IT-YOURSELF"
1540 PRINT "{down}_4)_I'M DONE!"
1600 PRINT :N=4:GOSUB 9000:IF T=0 THEN T=4
```

```

1610 PRINT "{clr}";:ON T GOTO 2000,3000,4000,8000
1800 GOTO 3000
1900 PRINT "{clr}"
2000 GOSUB 10000:GOSUB 11000:GOSUB 12000
2010 ROW=FN R(24-HT)-1:COL=FN R(39-WD)
2020 GOSUB 14000
2030 GET T$:IF T$=CR$ THEN 1500
2040 GOTO 2000
3000 PRINT "{clr}"
3010 FOR RW=0 TO 20 STEP 8
3020 FOR COL=0 TO 35 STEP 7
3030 GOSUB 10000:GOSUB 11000:GOSUB 12000
3040 ROW=RW:GOSUB 14000
3050 GET T$:IF T$=CR$ THEN 1500
3060 NEXT COL:NEXT RW
3070 GET T$:IF T$="" THEN 3070
3080 IF T$=CR$ THEN 1500
3090 GOTO 3000
4000 CH$="{clr 2°down}YOUR_CHOICE_OF_":EO=-1
4010 J=1:PRINT CH$;"EYEBROWS_ IS{down}"
4020 FOR I=1 TO AN
4025 IF EO>=0 AND (AW(I) AND 1)<>EO AND AW(I)<>0 THEN 4040
4030 PRINT J;"{left}:_";A$(I);"{down}":SV(J)=I:J=J+1
4040 NEXT I
4050 N=J-1:GOSUB 9000:IF T=0 THEN 1500
4060 AT=SV(T)
4070 IF EO<0 AND AW(AT)>0 THEN EO=AW(AT) AND 1
4110 J=1:PRINT CH$;"EYES_ IS{down}"
4120 FOR I=1 TO BN
4125 IF EO>=0 AND (BW(I) AND 1)<>EO AND BW(I)<>0 THEN 4140
4130 PRINT J;"{left}:_";B$(I);"{down}":SV(J)=I:J=J+1
4140 NEXT I
4150 N=J-1:GOSUB 9000:IF T=0 THEN 1500
4160 BT=SV(T)
4170 IF EO<0 AND BW(BT)>0 THEN EO=BW(BT) AND 1
4210 J=1:PRINT CH$;"NOSES_ IS{down}"
4220 FOR I=1 TO CN
4225 IF EO>=0 AND (CW(I) AND 1)<>EO AND CW(I)<>0 THEN 4240
4230 PRINT J;"{left}:_";C$(I);"{down}":SV(J)=I:J=J+1
4240 NEXT I
4250 N=J-1:GOSUB 9000:IF T=0 THEN 1500
4260 CT=SV(T)
4270 IF EO<0 AND CW(CT)>0 THEN EO=CW(CT) AND 1
4310 J=1:PRINT CH$;"MOUTHS_ IS{down}"
4320 FOR I=1 TO DN
4325 IF EO>=0 AND (DW(I) AND 1)<>EO AND DW(I)<>0 THEN 4340
4330 :PRINT J;"{left}:_";D$(I);"{down}":SV(J)=I:J=J+1
4340 NEXT I
4350 N=J-1:GOSUB 9000:IF T=0 THEN 1500
4360 DT=SV(T)
4370 IF EO<0 AND DW(DT)>0 THEN EO=DW(DT) AND 1

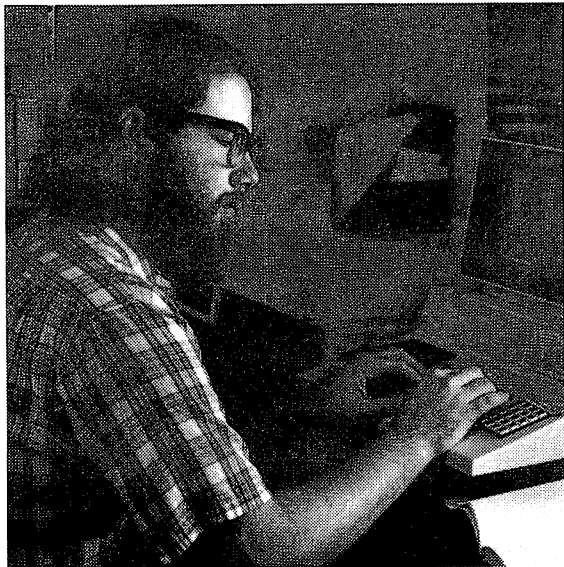
```

```

60120 IF Z=20 AND ZL>1 THEN IN$=LEFT$(IN$,ZL-1):PRINT "{left}";:
      GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>1)):FOR Z=2 TO ZL:PRINT Z$;:
      NEXT Z:GOTO 60000
60140 GOTO 60010
60500 FOR I=1 TO 13:PRINT "{DDD}";:NEXT I:PRINT :RETURN
62000 PRINT "{clr 2°down}";TAB(14-LEN(PG$));PG$;"_..BY..GLEN..FISHER"
62010 PRINT
      "{down}.....CURSOR.#5..COPYRIGHT.(C).1981.....{..}"
62011 GOSUB 60500
62020 PRINT "{down}YOU_(AND_YOUR_PET)_DRAW_FUNNY_FACES!"
62025 CR$=CHR$(13)
62030 PRINT "{2°down}PRESS_{rvs}RETURN{off}..TO..BEGIN";
62040 GET B$:IF B$<>" " THEN PRINT "{clr}":RETURN
62050 GOTO 62040

```

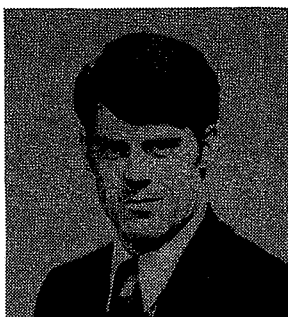
ABOUT THE AUTHORS



GLEN FISHER has been the editorial director of **CURSOR** since the first issue. In addition to his interest in computing, he is an active member of the Society for Creative Anachronism, and is an accomplished calligrapher.



RON JEFFRIES is the publisher of **CURSOR**, and president of The Code Works. When he isn't engrossed in designing new software products, he likes to read about computer science and electronics.



HOWARD ARRINGTON is a development engineer with Hewlett Packard, where he designs microprocessor hardware and software. He also owns Arrington Software Service which markets programs for the Exidy Sorcerer.



JULIA HALLFORD got interested in computers when her husband brought a PET home, "ostensibly for his work." She says that she became addicted, and became the primary user of the PET in their home.



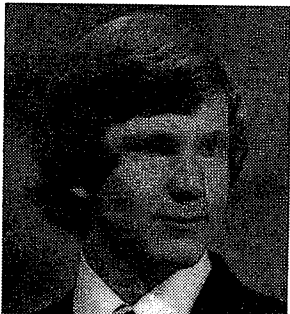
ART CARPET is a communications technician by day, and an avid computer software developer by night. He says, "I am overweight because I sit in front of my computer too much. However, my fingers are in good shape."



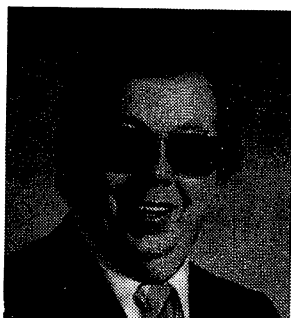
JENNI JEFFRIES is a student at Dos Pueblos High School, and has illustrated several school publications. She also enjoys raising puppies for Guide Dogs for the Blind in San Rafael, California.



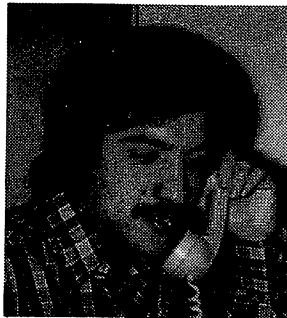
RANDALL LOCKWOOD is an assistant professor of psychology at the State University of New York, Stony Brook. Most of his research is on animal behavior, particularly wolves, bats, and rats. His most interesting computer application has been to use the PET to synthesize monkey calls. This was done to help keep a marmoset contented while it was hand-raised in an incubator next to the computer. He enjoys film-making, writing science fiction, and writing music.



TOM MARAZITA has worked for CLOAD Magazine as assistant editor for the past four years. He plans to major in electrical engineering or computer science when he attends college. When he isn't designing new boards for his homebrew 8080 computer, he enjoys assembly language programming, listening to short wave radio, and reading science fiction.



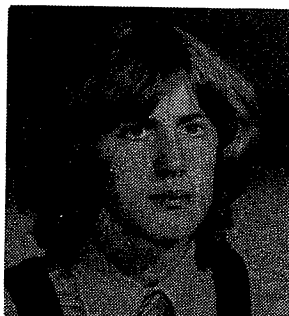
JOHN MATARELLA is the assistant principal of Arundel Middle School in Odenton, Maryland. He uses computers in his work for record-keeping, statistical analysis, and word processing.



JOHN MELISSA is a certified public accountant at Las Vegas, Nevada. When he wrote his contribution to the book, he was an agent with the Nevada State Gaming Control Board. He became so involved with the PET computer that he started a Commodore dealership at Las Vegas.



MALCOLM MICHAEL is a programmer for a hospital information systems business in San Mateo, California. He first got involved with computers when he used a PDP 8/e for a high school project. He says that he's forgotten the project, but that the exposure to computers changed his life. He has named his PET "Winthrop" and says that it is used almost exclusively for entertainment.



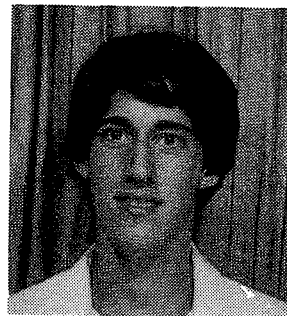
KEN MORLEY is an analytical chemist in Vancouver, British Columbia. He enjoys finding ways to use the graphic capabilities of the PET, as well as using it in chemistry. Other than computer programming, he enjoys photography, skiing, and hiking.



CHRIS NADOVICH works for Zeger-Abrams, a small firm that specializes in innovative electronic design. He was first introduced to computers in high school, where he learned BASIC and APL. In his spare time, he enjoys reading Spinoza, Russell, and Stephen King, as well as playing music and cooking.



DAVID PLATTON is a systems programmer for the Santa Barbara (California) Mental Health Department. At age 13, he got together with some friends to learn about computers and formed the West Coast chapter of R.E.S.I.S.T.O.R.S. (Radically Emphatic Students Interested in Science, Technology, and Other Research Studies).



BRIAN SAWYER is a student at Santa Barbara High School. He has been an assistant editor of CURSOR for two years. He learned computer programming when he was 12, and got his own PET when he was 14. When he isn't programming, he enjoys experimenting with digital electronics, playing electric guitar in a band, and making films.

Other OSBORNE/McGraw-Hill Publications

An Introduction to Microcomputers: Volume 0 — The Beginner's Book
An Introduction to Microcomputers: Volume 1 — Basic Concepts, 2nd Edition
An Introduction to Microcomputers: Volume 3 — Some Real Support Devices
Osborne 4 & 8-Bit Microprocessor Handbook
Osborne 16-Bit Microprocessor Handbook
8089 I/O Processor Handbook
CRT Controller Handbook
68000 Microprocessor Handbook
8080A/8085 Assembly Language Programming
6800 Assembly Language Programming
Z80 Assembly Language Programming
6502 Assembly Language Programming
Z8000 Assembly Language Programming
6809 Assembly Language Programming
Running Wild — The Next Industrial Revolution
The 8086 Book
PET[™] and the IEEE 488 Bus (GPIB)
PET[™]/CBM[™] Personal Computer Guide, 2nd Edition
Business System Buyer's Guide
Osborne CP/M[®] User Guide
Apple II[®] User's Guide
Microprocessors for Measurement and Control
Some Common BASIC Programs
Some Common BASIC Programs — PET[™]/CBM[™] Edition
Some Common BASIC Programs — Atari[®] Edition
Some Common BASIC Programs — TRS-80[™] Level II Edition
Some Common BASIC Programs — Apple II[®] Edition
Practical BASIC Programs
Practical BASIC Programs — TRS-80[™] Level II Edition
Practical BASIC Programs — Apple II[®] Edition
Payroll with Cost Accounting
Accounts Payable and Accounts Receivable
General Ledger
8080 Programming for Logic Design
6800 Programming for Logic Design
Z80 Programming for Logic Design
CBASIC[™] User Guide
Science & Engineering Programs — Apple II[®] Edition
Interfacing to S-100/IEEE 696 Microcomputer
A User Guide to the UNIX[™] System

PETTM Fun and Games

Climb the world's tallest mountain, battle a vicious monster, dare the odds with a roll of the dice, or venture through a maze from within. A wonderland of challenge and excitement await you and your PET computer.

Packed within these pages are programs for more than 30 fun-filled games and puzzles, selected as the best from more than three year's worth of games originally published by CURSOR cassette magazine.

All of the programs will run on any model Commodore PET or CBM personal computer.



PETTM is a trademark of Commodore Business Machines, Inc.
CURSORTM is a trademark of The Code Works.

ISBN 0-931988-70-5

